Knowledge-Based Systems 337 (2026) 115403

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

SORA-ATMAS: Adaptive trust management and multi-LLM aligned governance for future smart cities

Usama Antuley [a], Shahbaz Siddiqui [a], Sufian Hameed [a], Waqas Arif [a], Syed Attique Shah [b,*]

[a] Department of Computer Science, National University of Computer & Emerging Sciences, St-4 Sector 17-D On National Highway, Karachi, 75160, Pakistan
[b] Department of Computer Science, Birmingham City University, STEAMhouse, Belmont Row, Birmingham, B4 7RQ, United Kingdom

ARTICLE INFO

Keywords:
Adaptive trust management
Collaborative services
Agentic AI
GRC
Multi-LLM evaluation
Smart contracts
Smart-city governance
SDIoT

ABSTRACT

The rapid evolution of smart cities relies on agentic AI for autonomous decision-making, yet introduces governance, risk, and compliance (GRC) challenges in decentralized environments. We propose SORA-ATMAS, an adaptive trust management and multi-LLM governance framework for smart-city disaster management. Evaluation with Weather, Traffic, and Safety agents shows the framework steers multiple LLMs (GPT, Grok, DeepSeek) toward policy-aligned outputs, reducing mean absolute error by 35% on average. Results demonstrate stable weather monitoring, effective handling of high-risk traffic plateaus ($R \approx 0.85$), and adaptive trust regulation in safety scenarios. Runtime profiling confirms scalability, with throughput of 13.8–17.2 req/s, execution times < 72 ms, and governance delays < 100 ms for a 3-agent deployment; analytical projections indicate maintained performance at larger scales. Cross-domain policies ensure safe interoperability, such as allowing traffic rerouting only under validated weather conditions. SORA-ATMAS thus provides a regulation-aligned, verifiable governance framework that transforms distributed agent outputs into accountable, real-time decisions, offering a resilient foundation for smart-city management.

1. Introduction

The rapid emergence in smart cities has led to greater reliance on interconnected, intelligent services that enhance urban infrastructure, resource management and well-being of citizens. Specifically, these services utilize technologies such as the Internet of Things (IoT), big data, and artificial intelligence (AI) to establish efficient urban ecosystems [1,2]. As a result, smart cities address challenges related to city population, energy consumption, traffic density and service delivery by promoting sustainable and adaptive urban environments through real-time data exchange and process automation[3,4].

Agentic AI plays a vital role in urban development. It enables autonomous decision making and adaptive coordination, allowing systems to respond proactively to changing urban conditions [5,6]. In contrast to traditional AI, agentic systems can set goals, interact with their environments and collaborate independently. This makes them particularly effective for managing the complexities of smart city operations [7]. Such autonomy is essential for processing dynamic data from multiple sources. As a result, efficiency and scalability improve in smart urban environments[8]. For example, agentic AI can impact the energy sector by intelligently controlling consumption and forecasting future demands [9]. It can also improve the efficiency of renewable energy sources by analyzing historical usage data and weather patterns. In the transport sector, agentic AI can improve routing efficiency and reduce delivery times, and strengthen supply chain operations by continuously analyzing traffic patterns, weather conditions and real-time vehicle data[10].

The deployment of agentic AI in smart cities raises significant governance, risk, and compliance (GRC) challenges [11]. Effective governance requires clear accountability for autonomous decisions across interconnected urban services. Opaque decision-making can create ethical concerns and reduce transparency [12]. For example, in a smart-city traffic system without governance oversight, an autonomous AI agent may misinterpret camera data during heavy rainfall and infer a major traffic accident. Without validation or approval mechanisms, the agent may alter traffic signals and reroute vehicles citywide. This can cause severe congestion, delayed ambulance response, and disruption of public transportation. The core risk lies not in AI autonomy itself, but in the absence of control, verification, and accountability mechanisms [13]. Such failures can undermine system integrity and increase the risk of data breaches [14,15]. Weak governance can also erode public trust and hinder AI adoption. Recent studies emphasize the need for strong surveillance to prevent social harm [16].

* Corresponding author.
E-mail address: syedattique.shah@bcu.ac.uk (S.A. Shah).

https://doi.org/10.1016/j.knosys.2026.115403
Received 27 October 2025; Received in revised form 5 January 2026; Accepted 22 January 2026
Available online 2 February 2026

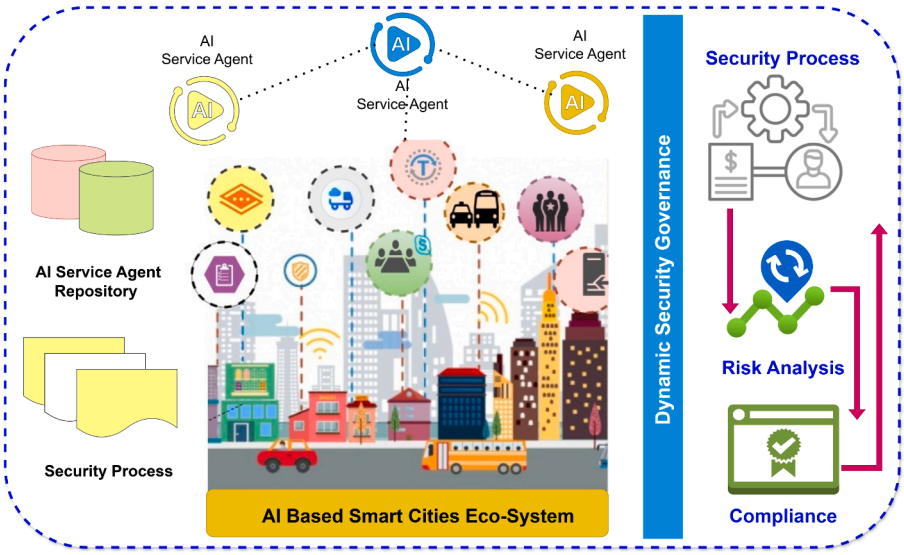

Fig. 1. An AI-driven smart city ecosystem. Illustration of how heterogeneous AI service agents operate across urban domains and highlights the need for coordinated governance, dynamic risk analysis, and compliance enforcement to ensure reliable and safe city-wide decision-making.

Blockchain is a key enabler of governance, risk, and compliance (GRC) in smart-city ecosystems due to its transparency, immutability, and decentralization [17]. In agentic AI systems, autonomous agents make decisions and exchange data across heterogeneous domains. Blockchain provides a trusted, tamper-proof record of actions, policies, and outcomes. Immutable logging supports auditing, accountability, and regulatory verification, which are essential for public trust [18,19]. Decentralization reduces reliance on single authorities and mitigates points of failure while ensuring consistent rule enforcement. As agentic AI systems scale, blockchain-enabled GRC becomes critical for transparency, traceability, and compliance in safety-critical smart-city operations [20].

Addressing these challenges requires an integrated approach that dynamically assesses risk and trust across smart-city domains. Such a system should combine contextual assessment of agent decisions with coordinated governance mechanisms and decentralized technologies such as blockchain. This integration enables secure and transparent data exchange. By embedding accountability, adaptability, and resilience into smart-city infrastructures, future frameworks can better manage GRC while supporting reliable and trustworthy agentic AI operations. Fig. 1 illustrates the interaction between GRC principles and smart-city domains.

Motivation

The integration of Agentic AI within the smart city ecosystem presents significant opportunities to optimize various urban services, including transportation, energy management, healthcare, and public safety [10,12]. By autonomously coordinating data from IoT devices, cloud infrastructures, and interconnected urban platforms, AI agents can significantly enhance operational efficiency, adaptability, and resilience across diverse smart domains [6]. While AI agents can autonomously interpret sensor data and coordinate actions across transportation, energy, healthcare, and public safety systems, their reliance on imperfect, incomplete, or adversarially manipulated data creates a significant risk of erroneous or hallucinated decisions[11,14]. For example, a traffic management agent may hallucinate severe congestion due to misinterpreted or spoofed sensor readings and autonomously reroute vehicles, inadvertently worsening congestion or obstructing emergency routes. Similarly, a safety agent analyzing CCTV streams could falsely infer a fire or hazardous event from visual noise or ambiguous cues, triggering unnecessary evacuations or emergency dispatches. In energy or healthcare domains, hallucinated demand spikes or misinterpreted patient vitals could lead to resource misallocation with direct societal consequences. Such erroneous decisions are particularly dangerous in smart cities because they can propagate rapidly across interconnected urban services, amplifying localized errors into city-scale disruptions.

Traditional governance mechanisms often fall short in managing these emerging challenges [21]. Therefore, ensuring trustworthiness, transparency, and regulatory compliance in Agentic AI necessitates adaptive trust management frameworks, blockchain-based governance models, and real-time output validation mechanisms [17,19].

1.1. Problem statement

Despite the growing deployment of Agentic AI in smart-city ecosystems, existing systems lack a formal governance foundation to ensure that autonomous decisions remain trustworthy, verifiable, and regulation-compliant under dynamic and high-risk conditions. Current solutions rely either on centralized control, which suffers from scalability and single-point-of-failure issues, or on decentralized agents with limited global oversight. As a result, erroneous or hallucinated outputs caused by noisy sensors, incomplete data, or misaligned reasoning can propagate across interconnected urban services and trigger cascading failures.

Formally, consider a smart-city ecosystem consisting of a set of autonomous agents

$$\mathcal{A} = \{A_1, A_2, \ldots, A_n\}, \tag{1}$$

where each agent A_i produces a decision $d_i(t)$ at time t, together with an estimated risk $R_i(t)$ and trust score $T_i(t)$ derived from local observations and reasoning models (e.g., LLMs). In many existing systems, decisions are validated independently using local thresholds, such that

$$\forall A_i \in \mathcal{A}, \quad d_i(t) \text{ is accepted if } T_i(t) \geq \tau_i, \tag{2}$$

without validation against global safety constraints, cross-domain dependencies, or regulatory requirements. This local validation allows incorrect or hallucinated outputs to satisfy confidence thresholds while violating city-level policies. Existing governance mechanisms also lack adaptive control over agent autonomy as contextual risk increases. They do not provide verifiable accountability or auditability in decentralized settings. Consequently, the system provides no formal assurance that the aggregated system state

$$S(t) = \{d_1(t), d_2(t), \ldots, d_n(t)\} \tag{3}$$

remains aligned with public safety goals, regulatory compliance, and trust expectations, particularly under emergency or high-uncertainty conditions, where $S(t)$ denotes the aggregated global system state and each $d_i(t)$ represents a decentralized decision independently produced by agent i at time t.

1.1.1. Research hypothesis

This work hypothesizes that trustworthy and regulation autonomy in smart city Agentic AI systems can be achieved by integrating continuous governance-level validation of agent outputs, adaptive trust and risk regulation across heterogeneous agents, and verifiable accountability through decentralized yet auditable governance mechanisms.

1.1.2. Research questions

Guided by the formulation and hypothesis of the above problem, this study investigates the following research questions.

1. *RQ1:* How can Agentic AI systems in smart cities be continuously monitored and validated to ensure accuracy, reliability, and fairness across heterogeneous domains?
2. *RQ2:* How can blockchain-based governance mechanisms establish decentralized transparency, accountability, and auditability for multi-agent smart-city ecosystems?
3. *RQ3:* How can adaptive trust management and specialized orchestration agents dynamically regulate agent autonomy and coordination to enhance resilience and regulatory compliance under high-risk conditions?

These research questions directly address the identified shortcomings of existing agentic smart-city systems and define the scope and objectives of the proposed SORA-ATMAS framework.

Contribution

The major contributions of this paper are outlined below, emphasizing both the novelty of the framework and its empirical validation against relevant baselines:

1. We propose SORA-ATMAS, a multi-agent adaptive trust management framework that integrates decentralized agentic sensing and reasoning at the edge with centralized governance validation through SORA. The framework employs a dual chain blockchain architecture Agentic Block
chains at the edge and a SORA-Blockchain at the governance layer to enable real-time, context-aware risk and trust evaluation while ensuring tamper-resistance, transparent compliance, verifiable cross-domain coordination, and city-level accountability without single points of failure.
2. To overcome the limitations of single-model systems, we design a multi-LLM evaluation mechanism. It integrates GPT, DeepSeek, and Grok as agent reasoning engines. Unlike online shielding [22] or static XACML policies [23], our ensemble combines MAE-based selection, governance thresholds, and iterative error feedback. This enables adaptive convergence across domains, achieving up to a 35% reduction in MAE across agents.
3. SORA-ATMAS is validated in a smart-city disaster management scenario with weather, traffic, and safety agents. The framework consistently converges towards the SORA baseline. It demonstrates robustness under high-risk conditions, such as traffic plateaus with $R \approx 0.85$. This ensures strong cross-agent interoperability and establishes scalability and reliability for diverse smart-city ecosystems.

The remainder of this paper is structured as follows. Section 2 discusses adaptive governance, risk, and compliance (GRC) challenges for agentic AI in smart cities. Section 3 presents the literature review. Section 4 describes the overall architecture of the proposed SORA-ATMAS framework. Section 5 introduces a decentralized and trusted GRC use case for smart city disaster management. Section 6 explains workflow execution. Section 7 presents experimental analysis, results, and discussion. Section 8 provides complexity analysis and addresses the research questions. Section 9 outlines the limitations of the proposed approach. Section 10 concludes the paper and highlights future research directions.

2. Adaptive governance, risk & compliance for AI

This section focuses on Governance, Risk, and Compliance (GRC) for Agentic AI in Smart Cities. Responsible use of AI in smart cities is dependent upon the correct implementation of GRC frameworks that allow for the appropriate alignment with regulations. In this study, we will be focusing on centralized GRC models and their limitations of security as well as their potential for improvement, including decentralization, accountability, trust management, and agentic governance models such as SORA. Centralized GRC allows the integration of city-based services with regulators and other local authorities, and provides a framework of regulation to govern areas of service such as energy, transportation, and surveillance through coordinated oversight [13]. Systems reliant on institutional monitoring, dashboards, audits, and enforcement to ensure compliance [24]. The benefits of using centralized governance include ensuring consistent policy, harmonizing regulations, and providing clear oversight of multiple and varied systems [25], while also allowing for the effective management of risk and the structured deployment of AI in urban planning.

2.1. Security challenges in centralized GRC for AI

Centralized GRC has its advantages in terms of offering consistent standards across all parts of a smart city ecosystem, but also creates many serious security risks for these types of ecosystems:

1. Centralized governance can create a bottleneck. Failures caused by system faults or cyberattacks (e.g, DoS) can propagate across interconnected urban services [15,26].
2. Scalability remains a major limitation. Continuous data streams from IoT devices and agentic AI systems overwhelm centralized risk evaluation, thereby constraining real-time adaptability and leaving cities vulnerable to emerging risks [19].
3. Centralized architectures amplify security and privacy risks [13]. Adversaries may spoof sensor data or exploit LLM prompt injection to distort risk assessments and manipulate agent decisions [27]. Central data repositories also increase exposure to large-scale breaches and unauthorized access, undermining trust, privacy, and regulatory compliance [28].

These challenges underscore why recent critiques argue that centralized GRC cannot meet the flexibility, resilience, and accountability requirements of heterogeneous multi-agent smart-city ecosystems [21].

2.2. Security requirements for adaptive GRC

In order to overcome the limitations, it is important to move from a rigid and centralized approach to a more flexible governance system. This system should include decentralization and distributed resilience while still allowing for central oversight when necessary. The following requirements address each of the challenges discussed.

1. Decentralized oversight reduces dependence on a single governance hub and improves resilience against denial-of-service attacks. Distributing compliance functions across multiple nodes allows orchestration agents such as SORA to maintain governance continuity under targeted disruptions [17,21,29].
2. Governance mechanisms must handle high-volume IoT and AI data while validating semantic reasoning outputs. Context-aware and flexible policy frameworks support local adaptation and prevent

Table 1
Coverage of reviewed studies across smart-city domains, Agentic AI integration, trust, interoperability, and governance dimensions.

Study	Smart City Domain	Agentic AI	Trust & Risk	Security & Compliance	Service Interoperability	Governance
Wang et al. [33]	Traffic Control	✓	×	×	×	×
Cai and Wei [34]	Traffic Signal Optimization	✓	×	×	×	✓
Hameed et al. [35]	Air Quality & Traffic Analytics	✓	✓	×	×	×
Elamanov et al. [36]	IoT / Industrial Services	✓	×	✓	✓	×
Lee et al. [37]	IoT and Edge Platforms	✓	×	✓	✓	✓
Siddiqui et al. [21]	Municipal Cross-domain Services	×	✓	✓	✓	✓
Siddiqui et al. [19]	Security Governance	×	✓	✓	✓	×
Islam et al. [17]	Blockchain-based Trust Systems	×	✓	✓	×	×
Ayub and Alshawa [32]	Smart City IoT Security	×	✓	✓	×	✓
Praharaj [38]	Integrated Command and Control Centers	×	×	✓	×	✓
Sohail et al. [39]	Urban Digital Twins	✓	×	×	✓	×
Rosmaninho et al. [40]	Edge–Cloud Orchestration	✓	×	✓	✓	✓
Our Framework	Multiple Smart City Domains	✓	✓	✓	✓	✓

rigid enforcement [30,31]. In the proposed framework, MAE-based validation and iterative feedback loops mitigate risks such as LLM prompt injection and prevent manipulated outputs from bypassing policy constraints.

3. Privacy and accountability must be strengthened through secure audit trails, distributed data storage, strict access control, and clear responsibility mapping [32]. These measures reduce large-scale breach risks, improve regulatory compliance, and enhance transparency in AI-driven decisions, supporting public trust [19].

Collectively, these requirements define an adaptive hybrid GRC architecture. The model balances centralized accountability with decentralized resilience. It also enables transparent and secure governance of smart city infrastructures, while at the same time providing operational robustness and promoting Public Trust.

3. Literature review

Smart cities are increasingly framed as cyber-physical ecosystems. Here, sensing, connectivity, and AI coordinate resources and services at the urban scale [29]. Empirical studies show how deep reinforcement-learning pipelines reshape urban operations. For example, city-scale traffic signal optimization with multi-agent RL controllers compresses travel times and reduces congestion [33,41,42]. Multimodal prediction models that fuse air quality, meteorological, and traffic data advance public-health and climate agendas [35]. Autonomous multi-agent systems, such as decentralized rendezvous planning for robots, boost urban functionality. These support operations include search and rescue, surveillance, and assembly [43]. Collectively, these results show that AI-enhanced programs lower lifecycle costs through predictive maintenance, streamline mobility by coordinating heterogeneous flows, and support sustainability targets with precise forecasting.

Building on assistive gains, research now pivots toward agentic AI systems that act with minimal supervision, often as multi-agent systems (MAS) [42]. In operational domains, MARL coordinates diverse services in real time. Agents negotiate priorities at distributed intersections, allocate resources and trigger actions under uncertainty [34]. Architecturally, edge-cloud designs push computation close to sensors and vehicles for low-latency, safety-critical tasks. Cloud layers synchronize global state and policy. Standardized M2M-MEC interworking and Modbus IoT gateways demonstrate interoperability with legacy and modern services [36,37]. These patterns boost responsiveness but complicate enforcement and explainability, motivating the adoption of structured GRC frameworks. Agentic models now explore counterfactual scenarios for proactive planning [34]. This highlights adaptability and pose governance challenges.

As cities move towards autonomy, GRC emerges as key enabler of trustworthy operations. Smart contract controls with blockchain logging support authentication, authorization and accountability across services [21], though scalability and policy rigidity remain concerns in large ecosystems. Adaptive governance extends these mechanisms with on-chain policy compilation and runtime compliance metrics [19], but incurs significant overhead in multi-domain settings. Decentralized trust frameworks that integrate blockchain with AI-driven techniques improve decision-making in safety-critical environments [17], yet introduce latency, energy costs, and a lack of regulatory pathways. Online shielding provides embedded compliance by blocking unsafe actions during learning and execution [44], though adaptability across complex MAS is limited. Centralized approaches include adaptive eXtensible Access Control Markup Language (XACML) for runtime-aware IoT access control [45], although increasing policy complexity reduces system robustness. In addition, integrated risk management (IRM) and GRC suites combined with blockchain and AI analytics enhance resilience [32,46]. However, high implementation costs, vendor dependency and scalability limit widespread adoption. At the urban scale, Integrated City Command and Control Centers (ICCCs) illustrate both the potential benefits and inherent risks of centralized governance. These centers enhance administrative oversight but also introduce vulnerabilities due to corporate influence, inconsistent deployment and insufficient mechanisms for public accountability [38]. These challenges are exacerbated by interoperability. Smart city infrastructure, vendor-specific protocols, and numerous fragmented schemas all contribute to the inefficiencies of smart cities, resulting in bottlenecks in cross-domain coordination. Interoperable standards-based Edge-Cloud communication [36,37] and Digital Twins (DTs) with real-time environmental, mobility, and utility data, provide a common substrate for making decisions [39]. However, unified standards-based centralized hubs for unifying standards can act as throughput bottlenecks and create latency in responding to events; thus, adaptive governance. Adaptive governance distributes authority across district, utilities, and agent levels while preserving citywide policy objectives.

Despite recent progress, two gaps continue to limit agentic smart-city systems. First, the link between explainability and governance remains weak. Existing methods rarely provide verifiable evidence that autonomous decisions comply with legal, ethical, and operational requirements. Runtime enforcement through online shielding is a promising step, but its use is still limited and has not been generalized to multi-agent platforms [44]. Additionally, achieving equity and resilience in cities requires advancements in both institutions and technology. Prior studies have emphasized that standardized edge-cloud orchestration is required for reliable coordination of critical urban services [40]. Similarly, transparent digital twin pipeline structures will be needed to support the observability of urban dynamic processes [39] and to provide trust frameworks that provide quantifiable assurances about security and benefit sharing [17]. These findings collectively suggest a hybrid governance path; one in which centralized Governance Risk Compliance (GRC) provides both standardization and veto authority over high risk activities, while decentralized, context aware decision execution agents

operate on an equitable scale. Table 1 summarizes the coverage of these studies across agentic AI integration, interoperability, and governance models.

4. System overview

The proposed framework is a layer-based using a Software-Defined Internet of Things (SDIoT) structure to provide a structured environment for adaptive agent-based decision-making in smart cities (See Fig. 2). SDIoT is treated as a enabling abstraction as opposed to being the main contribution; it allows modularity, programmability, and scalability for distributed coordination in heterogeneous IoT environments. This structure also enables the separation of perception, application, and control layers with the ability to perform trust assessment at an integrated level, enforcing decentralized policies, and enabling accountability at a city level.

1. *Application layer*: At the top, the Application Layer is combined with the SORA Governance Layer and the Agentic Layer. The SORA Governance Layer has the function of an oversight authority at a city level. It provides the engine for the dissemination of security policies, the coordination between domains, the assessment of adaptive trust and the enforcement of GRC, based on governance repositories using blockchain technology. The Agentic Layer in turn allows for autonomy specific to each domain via decentralized agents for weather, traffic, and public safety. These agents provide for compliance checks, contextual reasoning through LLMs and local blockchain anchoring. Collectively, both layers allow for autonomous services that are compliant to regulations, verifiable and contextually aware and yet can be adapted locally.
2. *Control layer*: Below the Application Layer, the Control Layer manages communication, heterogeneity and scalability via SDN-Inspired Mechanisms and will utilize multiple Controllers to enable Load Balancing and Fail-Over capabilities in addition to enabling automatic activation of backup Controllers in the event of a Failure. This Layer separates the Control Plane from the Data Plane which enables programmability and the ability to discover Topology and manage flows across all IoT Domains, and provides interoperability by handling packets through the WISE Flow Table for IEEE 802.15.4. In addition to providing Interoperability, the layer also handles Cryptographic Operations such as generating ECC Keys (128/192/256-bit), performing ECDH Session Key Exchange, and performing Policy-Defined Key Rotation utilizing Curves such as “secp256r1” and “secp384r1”. These operations enable authenticated and encrypted communication and provide the necessary infrastructure for Secure Policy Enforcement.
3. *Perception layer*: At the bottom, the perception layer consists of distributed IoT devices. These include environmental sensors (temperature, humidity, wind, rainfall), traffic monitoring systems, and safety sensors or cameras for fire and smoke detection. Devices capture real-time data that is pre-processed and logged into structured repositories. The collected data feeds trust evaluation, risk estimation, and governance decisions. Observations of weather-related events, traffic incidents, and safety factors contribute to real-time monitoring for emergency response, optimized travel routes, and public safety.

These three layers provide a structural foundation for the framework, as shown in Fig. 2. The design supports modularity and scalable integration of different types of IoT-based services while SORA governance and agentic intelligence promote adaptive trust and risk-aware compliance and resilience within cross-domain operations.

4.1. Proposed architecture

The proposed architecture implements a dual-chain governance model. Decentralized AI agents operate autonomously within their domains while remaining accountable to global city governance. Fig. 3 illustrates the integration of domain-specific AI agents, the Agentic Blockchain, and the SORA Governance Layer over the SORA Blockchain. This design aligns local intelligence and contextual adaptability with city-wide compliance, auditability, and interoperability.

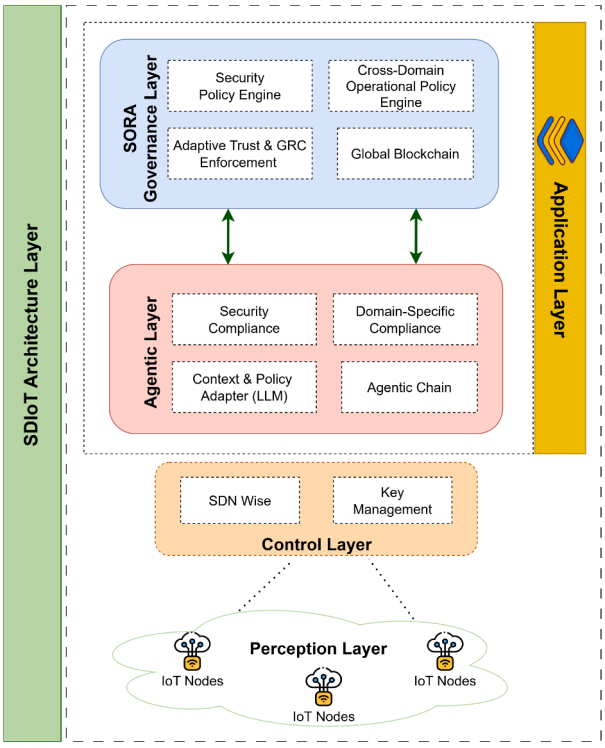

Fig. 2. The SDIoT Governance Framework as a layered model with SORA Governance Layer & Agentic Layer within the Application Layer, demonstrating functional separation for local autonomy at the perception, control, and agent level while the Global Layer provides uniform, risk-aware oversight and enforcement of policy across all City Services.

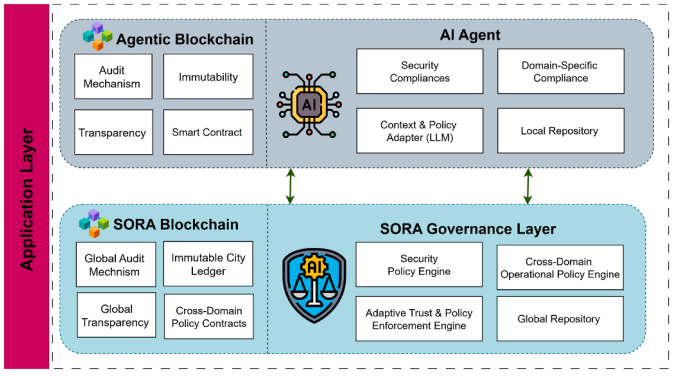

Fig. 3. The proposed Governance Framework integrates decentralized AI agents, Agentic Blockchain, and the SORA Governance Layer on top of the SORA Blockchain; validating the Dual Chain Design to show that agent-level decision-making is immutably recorded in each node’s blockchain, while the SORA Blockchain enforces global audits, cross-domain policies, and escalation logic.

4.2. Agentic layer

The AI Agents are located at the Application Layer, and they support a variety of smart-city services (i.e., transportation, health care, environmental protection, and public safety). Each Agent is autonomous in operation relative to the service domain, but each also remains ac-

countable for their respective operational activities to applicable laws and regulations that govern the city. Unlike static services, Agents contain both Compliance Intelligence and Adaptive Reasoning, which allow them to dynamically respond to changing contexts, newly emerging risks, or changes in policy.

Each AI agent integrates four interconnected components:

1. *Security compliance*: Authentication, authorization, encryption, and integrity validation are provided by this module. This ensures all exchanges of information between authentic Services and authenticated entities are secure. ECC-based Cryptography, along with SHA-256 integrity checks, prevents Tampering and unauthorized access.
2. *Domain-specific compliance*: The service domain's regulatory requirements are embedded within this module. For example, the health agents enforce GDPR. The safety thresholds for Real-Time Operations are enforced by the traffic agents. These regulations allow for lawful operation as well as optimization of functionality.
3. *Context and policy adapter (LLMs)*: This module uses large language models to interpret IoT streams, logs, and service requests. It evaluates semantic alignment with baseline policies. When anomalies occur, such as congestion, abnormal health indicators, or hazards, it recommends enforcement actions. These include access restriction, resource reallocation, encryption escalation, or alert generation. This capability enables adaptive and context-aware compliance.
4. *Local repository*: The local repository securely stores validated policies, compliance states, and trust-risk scores. Non-sensitive data is stored in conventional databases. Sensitive data, including PII from traffic or CCTV sources, is anonymized or removed after defined retention periods (e.g., 30 days). Techniques such as data aggregation, face masking, encryption, and role-based access control ensure privacy and regulatory compliance.

4.3. SORA Governance layer

AI agents provide local autonomy, but the *SORA Governance Layer* retains final authority. It acts as the city-level ruling entity for validation and approval of agent decisions. This ensures alignment with city-wide objectives, systemic risk thresholds, and regulatory mandates. Agents autonomously generate proposals, such as encryption escalation, access restriction, or domain-rule adaptation. These outputs are treated as recommendations. Formal approval is required before enforcement. This hierarchical structure preserves rapid local responsiveness while maintaining global accountability, trust, and regulatory compliance. Agent proposals are initially logged on their local Agentic Blockchain for provenance, while only SORA's final validation decisions are anchored on the SORA Blockchain for global auditability.

The SORA Governance Layer consists of four tightly integrated engines. Each engine performs a distinct supervisory role while operating as part of a unified governance workflow.

1. *Security policy engine:* The Security Policy Engine is responsible for defining global security baselines (cryptography standards, authentication factors, etc.) against which all security checks performed at the agent level (ECC Tier selection and ACL validation) are validated, to ensure that all agents provide a uniform level of protection.
2. *Cross-domain operational policy engine:* This engine ensures that each domain meets its own regulatory and operational requirements before interacting with other services. For example, healthcare agents must comply with GDPR, traffic systems must respect safety constraints, and energy services must follow sustainability policies. Once these domain-specific checks are satisfied, the engine manages cross-domain interactions by guiding how data is shared, how emergencies are coordinated, and how information is retained.
3. *Adaptive trust and risk enforcement engine:* This engine continuously monitors trust signals and aggregated risk across agents. It can override local trust assessments when thresholds are exceeded. Enforcement actions include access restriction, service suspension, or escalation to human oversight. This ensures that trust is globally enforced rather than locally asserted.
4. *Global repository:* This repository stores validated policies, agent registrations, risk-trust histories, and enforcement outcomes. It is anchored to the SORA Blockchain to provide immutable and auditable records. Stored knowledge is also fed back to agents to support adaptation and policy-aware learning.

Together, the Agentic Layer and the SORA Governance Layer form a coordinated dual-chain governance architecture. As shown in Figs. 2 and 3, data flows from the perception layer through the control layer to domain-specific agents. Agents evaluate context, assess compliance, and generate proposals that are logged locally and submitted to SORA. SORA validates decisions, enforces city-wide policies, and records approved actions for audit and accountability. Detailed execution flows are presented later through concrete use cases and algorithms (Sections 5.3 & 6. Operational validation follows Algorithm 1.

Algorithm 1 Policy validation and enforcement by SORA.

Input: Agent proposal $P_i = \langle S_i,\ \text{action},\ R_i,\ T_i \rangle$, local policy π_i, global thresholds (θ_R, θ_T), optional partner S_j

Output: Decision $\in \{\texttt{approve}, \texttt{restrict}, \texttt{deny}\}$ anchored on SORA-Chain

Security check (Security Policy Engine): Verify identity, authentication, authorization, and access control for S_i. If any fail → `deny`.
Domain-specific compliance: Confirm that S_i satisfies sectoral rules (e.g., GDPR confidentiality/minimization, mobility safety thresholds, energy sustainability/retention). If non-compliant → `deny`.
Trust/risk gate:
if $T_i < \theta_T$ **then**
    decision ← `deny` ▷ trust below threshold
else if $R_i > \theta_R$ **then**
    decision ← `restrict` ▷ limit scope or require monitoring/human review
else
    decision ← `approve`
end if
Cross-domain clause (Cross-Domain Operational Policy Engine): If proposal involves S_j, first ensure both agents pass their own domain-specific checks; then enforce cross-domain constraints (scope, data minimization, safety, retention). If any violation → decision ← `deny`.
Anchoring: Record $\{S_i, S_j, \text{action}, R_i, T_i, \text{decision}\}$ in `GovDecisions` and append an immutable entry to the SORA Blockchain.
return decision to S_i (and S_j if applicable).

4.4. Adaptive trust and GRC enforcement

To operationalize governance across heterogeneous smart-city services, we formalize how each *agent* quantifies (i) environmental risk, (ii) service reliability, (iii) contextual trust, and (iv) adaptive (overall) trust and risk, and how the governance node (SORA) enforces city-level policies with measurable tolerances. The constructs are domain-agnostic and apply to diverse smart-city services such as weather, mobility, utilities, health, and safety. The framework provides compact, auditable hooks from per-agent assessments to global orchestration and review.

Mathematical preliminaries. For all definitions, $\mathbb{I}(\cdot)$ denotes the indicator function:

$$\mathbb{I}(\text{condition}) = \begin{cases} 1, & \text{if the condition is true,} \\ 0, & \text{otherwise.} \end{cases}$$

All inputs are normalized to $[0, 1]$ prior to computation unless otherwise stated.

Definition 1 (Environmental risk). Environmental risk quantifies the extent to which real-time sensor or service observations deviate from expected operating conditions within a smart-city ecosystem. The formulation is domain-agnostic and applies to three generic data modalities: continuous signals, capacity or volume conditions, and discrete hazard events. It is defined as:

$$R^i_{\text{Env}}(t) = \begin{cases} \frac{1}{n}\sum_{k=1}^{n} \mathbb{I}\big(|x_k - \mu_k| > \theta_k\big), & \text{continuous signals,} \\ \mathbb{I}\big(\text{Load}(t) > \theta_{\text{cap}}\big), & \text{capacity/volume conditions,} \\ \mathbb{I}\big(\text{HazardEvents}(t) \geq 1\big), & \text{discrete hazard events.} \end{cases} \tag{4}$$

where n is the number of monitored environmental parameters; $x_k(t)$ is the normalized real-time observation of parameter k at time t; μ_k denotes the expected baseline value of parameter k; θ_k is the acceptable deviation threshold for parameter k; $\text{Load}(t) \in [0, 1]$ represents the normalized utilization level; $\theta_{\text{cap}} \in [0, 1]$ is the maximum safe capacity threshold; and $\text{HazardEvents}(t)$ denotes the count of verified hazard events within the evaluation window.
Typical triggers include abnormal climate measurements, excessive transport or infrastructure utilization, and verified public-safety or health-related hazard events.

Definition 2 (History-reputation trust; HRT). This measure captures the trustworthiness of an agent by combining its historical performance with peer reputation, while discounting older observations over time.

$$T^i_{\text{HRT}}(t) = \begin{cases} T_0, & t = t_0, \\ \delta\, T^i_{\text{HRT}}(t - \Delta T) + (1 - \delta)\big(\omega_p\, s(t) + \omega_r\, T^i_{\text{Rept}}(t)\big), & \text{otherwise,} \end{cases} \tag{5}$$

where $T_0 = 0.5$ is the initial trust value at time t_0; t denotes the current discrete trust evaluation instant; $\Delta T > 0$ is the fixed trust update interval with evaluations at $t = t_0 + k\Delta T$ for $k \in \mathbb{N}_0$; $\omega_p = .7$ and $\omega_r = 0.3$ are weighting coefficients satisfying $\omega_p + \omega_r = 1$; $\delta \in (0, 1)$ is the exponential forgetting factor; and $s(t) \in \{0, 1\}$ denotes the outcome of the most recent interaction observed over $(t - \Delta T,\, t]$. Typically, $\delta \in [0.8, 0.9]$ balances responsiveness to recent behavior with robustness against transient fluctuations, while the representative weighting $\omega_p = .7$ and $\omega_r = 0.3$ prioritizes direct experience over peer reputation without losing sensitivity to ecosystem consensus. These parameters are policy-configurable and calibratable according to service criticality and governance requirements.

Peer reputation trust derives from model alignment within the ecosystem:

$$T^i_{\text{Rept}}(t) = \frac{\sum_{j=1}^{M} c_j\, \mathbb{I}(f_j = f^\star)}{\sum_{j=1}^{M} c_j}, \qquad T^i_{\text{Rept}}(t_0) = 0.5. \tag{6}$$

where $f^\star$ denotes the consensus outcome among peers whose cumulative credibility exceeds a predefined governance threshold. Similarly, M is the number of peer agents, $c_j \in [0, 1]$ denotes the credibility of peer j, and $f_j \in \{0, 1\}$ is the feedback reported by peer j.

Definition 3 (Service risk). Service risk quantifies the likelihood of a service exhibiting unreliable behavior in the near future based on its past operational and reputational performance. Rather than relying solely on raw historical outcomes, service risk is defined as the complement of the history–reputation trust evaluated at the most recent trust update instant, thereby incorporating both direct interaction results and peer consensus.

$$R^i_{\text{Service}}(t) = 1 - T^i_{\text{HRT}}(t - \Delta T), \qquad R^i_{\text{Service}}(t_0) = 0.5. \tag{7}$$

This formulation captures direct performance risk through recent interaction outcomes and reputational risk through peer feedback embedded in T^i_{HRT}. Initializing $R^i_{\text{Service}}(t_0) = 0.5$ reflects a neutral risk assumption under maximum uncertainty, consistent with reliability engineering principles where failure probability is modeled as the complement of system reliability.

Definition 4 (Overall agent risk). The overall risk for each agent is determined by a combination of environment-driven anomalies and service reliability. A tunable parameter λ_i controls the relative contribution of environmental risk versus service risk.

$$R^i(t) = \lambda_i\, R^i_{\text{Env}}(t) + (1 - \lambda_i)\, R^i_{\text{Service}}(t), \tag{8}$$

where $\lambda_i \in [0, 1]$ is a domain-dependent weighting parameter. Since both $R^i_{\text{Env}}(t)$ and $R^i_{\text{Service}}(t)$ are normalized to $[0, 1]$, their convex combination guarantees that $R^i(t) \in [0, 1]$. The value of λ_i reflects the sensitivity of a given domain to environmental conditions. In practice, higher values are assigned to domains where environmental factors play a dominant role (e.g., public safety or health services), while lower values are used in domains where service reliability is the primary driver (e.g., software-centric information services). These weights are policy-configurable and can be calibrated using historical incident data or governance-driven risk assessments.

Definition 5 (Contextual trust). Contextual trust adjusts a baseline confidence level according to operational factors including data freshness, integrity and compliance. This mechanism enables agents to degrade performance in a controlled manner as operational conditions deteriorate.

$$T^i_{\text{Ctx}}(t) = \min\left(T_{\text{base}} \prod_{k=1}^{n_i} (M_{i,k}(t))^{w_{i,k}}, 1.0 \right), \tag{9}$$

where $T_{\text{base}} \in (0, 1)$ is the baseline trust assigned at service registration; n_i denotes the number of contextual modifiers for agent i; $M_{i,k}(t) \in [0, 1]$ is the k-th normalized contextual modifier at time t (e.g., data freshness, sensor integrity, protocol compliance); and $w_{i,k} \in [0, 1]$ are importance weights satisfying $\sum_{k=1}^{n_i} w_{i,k} = 1$. These parameters are policy-configurable and calibratable using historical behavior, domain-specific risk tolerance, or governance-defined thresholds; a representative choice (e.g., $T_{\text{base}} = 0.7$) reflects initial trust with sufficient margin for contextual degradation.

Definition 6 (Overall trust). Overall trust represents the aggregate trustworthiness of an agent by combining its historical–reputation trust and contextual trust using risk-adaptive weighting.

$$T^i_{\text{Overall}}(t) = w_{\text{HRT}}(t)\, T^i_{\text{HRT}}(t) + w_C(t)\, T^i_{\text{Ctx}}(t), \tag{10}$$

where

$$w_{\text{HRT}}(t) = 0.5 - 0.2\, R^i(t), \quad w_C(t) = 0.5 + 0.2\, R^i(t).$$

These weights satisfy $w_{\text{HRT}}(t) + w_C(t) = 1$ for all t, ensuring that $T^i_{\text{Overall}}(t) \in [0, 1]$. Since $R^i(t) \in [0, 1]$, the weights are bounded as $w_{\text{HRT}}(t) \in [0.3, 0.5]$ and $w_C(t) \in [0.5, 0.7]$, guaranteeing non-negative contributions. Reputation effects are incorporated exclusively through $T^i_{\text{HRT}}(t)$, avoiding double counting while preserving sensitivity to peer consensus. As agent-level risk increases, the model places greater emphasis on contextual trust to reflect heightened sensitivity to real-time operating conditions.

Definition 7 (Ecosystem metrics). At the city scale, ecosystem metrics aggregate trust and risk across all active agents to provide a global view of system reliability and safety.

$$T_{\text{Ecosystem}}(t) = \frac{1}{|\mathcal{A}(t)|} \sum_{i \in \mathcal{A}(t)} T^i_{\text{Overall}}(t),$$
$$R_{\text{Ecosystem}}(t) = \max_{i \in \mathcal{A}(t)} R^i(t). \tag{11}$$

The maximum operator used for ecosystem-level risk reflects a conservative governance strategy: the presence of a single high-risk agent is sufficient to trigger system-wide monitoring or corrective action. This choice is consistent with established practices in security- and safety-critical systems, where overall safety is constrained by the weakest component. In contrast, the average ecosystem trust value provides a representative measure of overall system performance and is useful for assessing service quality, planning capacity, and allocating resources.

GRC enforcement and threshold calibration. SORA enforces GRC by mapping continuous trust and risk metrics (Definitions 1–7) to explicit policy thresholds, as summarized in Table 3. These thresholds govern admission, escalation, access control, and intervention decisions. At the agent level, disaster-related actions are triggered when risk exceeds θ_R and trust falls below θ_T, i.e., $R^i(t) \geq \theta_R$ and $T^i_{\text{Overall}}(t) \leq \theta_T$. Representative values $\theta_R = 0.70$ and $\theta_T = 0.60$ balance early detection and false-positive suppression. Higher θ_R and lower θ_T reduce alert fatigue in non-critical domains, while safety-critical scenarios favor higher sensitivity. Trust thresholds also regulate access control. Highly trusted agents receive full privileges, moderately trusted agents operate under restrictions, and low-trust agents are denied access. Thresholds are tightened automatically during elevated ecosystem risk. At the governance layer, SORA enforces domain-specific admission thresholds $\tau_T(i)$ based on safety criticality, e.g., $\tau_T = 0.60$ for weather, 0.55 for traffic, and 0.65 for fire services. These values are calibrated using historical incidents, regulatory margins, and acceptable false-positive rates. Ecosystem-level enforcement uses aggregated metrics $R_{\text{Ecosystem}}(t)$ and $T_{\text{Ecosystem}}(t)$. City-wide threshold violations trigger coordinated escalation or human-in-the-loop review. Hysteresis and cooldown constraints (Table 3) prevent oscillatory behavior. Together, this threshold-driven strategy converts trust and risk scores into enforceable and auditable GRC controls.

4.5. Agentic and SORA blockchains

The governance framework uses a dual-chain setup for local and city-wide oversight. Each AI agent logs decisions on an Agentic Blockchain, a decentralized ledger for audits and domain rules via smart contracts. At the global level, the SORA Blockchain gathers evidence from all domain chains, provides cross-domain audits, enforces shared contracts, and manages trust, escalations, and emergency actions. Together, these chains form layered trust: Agentic Chains ensure local compliance, while the SORA Chain consolidates this into risk-aware, global governance. This structure avoids single points of failure and balances local flexibility with city goals.

In the current implementation, both the Agentic and SORA blockchains operate as permissioned MultiChain networks using the built-in Proof-of-Work (PoW) consensus with a fixed validator set. This configuration yields stable and predictable throughput suitable for governance logging, auditability, and policy enforcement rather than high-frequency transaction processing. Alternative consensus mechanisms (e.g., PoA or BFT-based protocols) can be adopted in future deployments to improve throughput without affecting the proposed trust–risk–GRC models.

The end-to-end enforcement flow that unifies per-agent execution and global validation is specified in Algorithm 2 (with metric computations referencing Definitions 1-7).

Algorithm 2 Policy enforcement and logging via dual-chain governance.

Input: Agent observation for S_i (`AgentLogs`); key K_b; thresholds (θ_R, θ_T); domain baselines $\tau_T(i)$; tolerances $(\epsilon_R, \epsilon_T, \epsilon_{\text{tie}})$; stability h; cooldown Δt; optional partner S_j. Metrics computed per Definitions (Defs) 1–7 in section 4.4 ; policy conditions from Table 3. **Output:** Decision $\in$ {`approve`, `restrict`, `deny`} anchored on Agent-Chain and SORA-Chain.

1: **Agent-side (W1–F1):** Authenticate K_b; preprocess input; compute R_{Env}, T_{HRT}, R_{Service}, R^i, T_{Ctx}, and T_{Overall} (Defs. 1–6). Append results to `AgentLogs` and anchor record on *Agent-Chain*. Forward packet $P_i = \{S_i, t, R^i, T^i_{\text{Overall}}\}$ to SORA.
2: **SORA ingress (S1):** Validate and admit only if packet is consistent within (ϵ_R, ϵ_T) and $T^i_{\text{Overall}} \geq \tau_T(i)$, following gate criteria in Table 3.
3: **Selection & feedback (S2–S3):** If multiple variants exist, select highest trust (tie → closest risk). Issue error-directed feedback $(\Delta R, \Delta T)$ to non-selected candidates, per policy rules.
4: **Decision (policy thresholds):** If $T_i < \theta_T \rightarrow$ `deny`; Else if $T_i <$ 0.7 or $R_i > \theta_R \rightarrow$ `restrict`; Else → `approve`. If partner S_j involved, enforce cross-domain rules; violations → `deny`.
5: **Anchoring/Logging:** Record $\{S_i, S_j, R_i, T_i, \text{decision}\}$ in `GovDecisions` and append to *SORA-Chain* Table 3, S2–S3).
6: **System-level metrics (S4–S6, Def. 7):** Compute ecosystem trust and risk. If ≥ 2 agents exceed high-risk triggers → `joint actuation` (S4). If ecosystem thresholds exceeded → `city-wide escalation` (S5), else apply trust-constrained escalation. Enforce hysteresis h and cooldown Δt to maintain stability (S6); anchor outcomes to SORA-Chain.

5. Use case description: decentralized, trusted GRC for smart-city disaster management

Building on prior work advocating hybrid governance that preserves city-level policy control while decentralizing execution, we implement a smart-city disaster management use case to balance agility and accountability. Agents operate near their data and apply defense-in-depth principles, while a lightweight governance layer enforces policy, provenance, and auditability without introducing centralized bottlenecks [19,47]. Three domain agents Weather, Traffic, and Safety operate under dual-chain governance. Agent outputs are signed on the *Agentic Blockchain*, while final governance decisions are anchored on the *SORA Blockchain*. All agents follow the unified orchestration and escalation workflow defined in Algorithm 2 and Table 3, as detailed in Section 6.

The roles, data sources, and processing pipelines of the three domain agents are summarized below.

1. *Weather agent*: The Weather Agent processes meteorological data for Karachi and Islamabad obtained from the Open-Meteo API for the period 2014–2025 as shown in (Fig. 4). The dataset contains approximately 196,000 records and includes temperature, precipitation, humidity, wind speed, cloud cover, UV index, and soil temperature. Missing values are removed ($< 1.2\%$), and features are normalized using min–max scaling. Operational regimes are labeled as *Heavy Rain* (≥ 40 mm/day), *Rain* (5–20 mm/day), and *Heatwave* (temperature ≥ 40°C or anomaly +5°C with UV ≥ 8); remaining conditions are labeled *Normal* [48,49]. A supervised XGBoost classifier predicts regimes and produces 20-hour forecasts. Outputs are stored as structured CSV files and consumed by three LLMs GPT-4.1 nano, Grok-4, and DeepSeek-R1 to generate trust and risk assessments, including predicted labels and explanatory reasoning. Signed outputs are anchored on the *Agentic Blockchain* and guide SORA's policy enforcement.
2. *Traffic agent*: The Traffic Agent estimates vehicular density as shown in (Fig. 5) from CCTV streams using a pre-trained YOLOv8 model designed for traffic density estimation [50]. The model is adopted directly from the publicly available implementation and is not retrained in this study. Vehicle density is computed as the number of detected vehicles per 100 m road segment, where each vehicle corresponds to a detected bounding box. A congestion threshold of 15 vehicles per 100 m is used, consistent with urban mobility studies [51]. Detection outputs include bounding boxes, confidence scores, and aggregate vehicle counts, which are logged in structured CSV files. These outputs are consumed by GPT-4.1 nano, Grok-4, and DeepSeek-R1 using fixed prompt templates to assess conges-

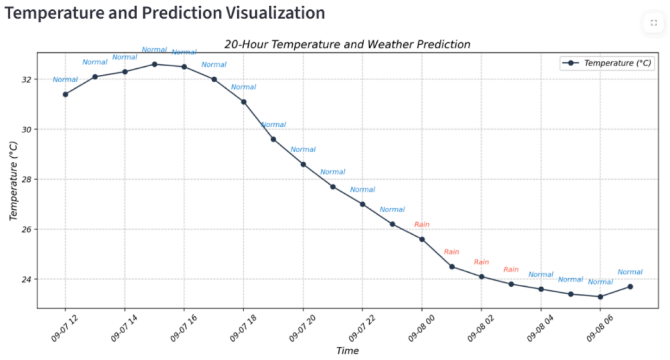

Fig. 4. 20-hour temperature and weather regime forecast (Islamabad example) shown via a Streamlit-based visualization. The line graph displays predicted temperature trends with overlaid regime labels (*Normal/Rain*). These regime classifications serve as contextual inputs to the Weather Agent, directly influencing environmental risk estimation and adaptive trust computation under changing weather conditions.

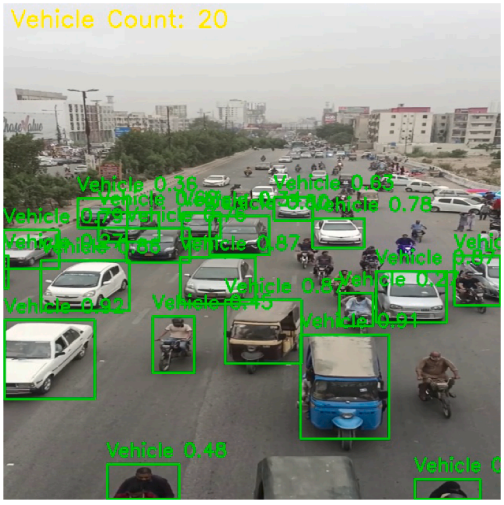

Fig. 5. YOLOv8 traffic-density output with bounding boxes and confidence scores. The detected vehicle counts and confidence levels are aggregated to quantify congestion intensity, which is then mapped to the Traffic Agent's risk and trust metrics for downstream governance and control decisions.

tion risk and to evaluate trust and risk with explanatory reasoning. Governance-approved recommendations, such as adaptive signaling or traffic diversion, are signed and anchored on the *Agentic Blockchain*.

3. *Safety (Fire/Smoke) agent*: The Safety Agent detects fire and smoke events in CCTV streams as shown in (Fig. 6) using a pre-trained YOLO11 "Flare Guard" model for real-time hazard detection [52]. The model is adopted as provided and is not fine-tuned further. Each detection is logged with a timestamp, hazard type (fire or smoke), confidence score, and spatial location. Detections with confidence scores ≥ 0.5 are forwarded to GPT-4.1 nano, Grok-4, and DeepSeek-R1 for contextual annotation and trust–risk evaluation with explanatory reasoning. Signed outputs are recorded on the *Agentic Blockchain*. Hazard confidence directly elevates environmental risk and triggers SORA's escalation and emergency response policies.

A consolidated summary of the data sources, model performance, primary outputs, and ledger anchoring for all agents and the governance authority is presented in Table 2.

Fig. 6. YOLO11 "Flare Guard" output on a hazard scene showing detected smoke (0.74) and fire (0.55) regions. These confidence scores are interpreted as hazard severity indicators and are used to elevate environmental risk, triggering SORA's escalation and emergency response policies in safety-critical scenarios.

5.1. Governance authority

SORA (Security & Operational Response Agent) acts as the supervisory governance node, maintaining its own trust-risk reference to evaluate each agent's LLM outputs. The model with the lowest mean absolute error (MAE) per interval is selected as authoritative, while others receive signed, policy-based feedback for convergence. All governance artifacts validations, selections, feedback, policy versions, and directives are immutably recorded on the *SORA Blockchain*, with the *Agentic Blockchain* ensuring edge provenance. This dual-ledger setup supports embedded compliance, human-on-the-loop oversight, and tamper-evident auditability in multi-agent smart-city systems [13,19,53].

5.2. Operational policy matrix

Governance enforcement in SORA builds on the trust and risk constructs defined in Section 4.4, operating on Overall Agent Risk $R^i(t)$ (Def. 4), Overall Trust $T^i_{\text{Overall}}(t)$ (Def. 6), and aggregated ecosystem metrics (Def. 7). The unified policy matrix (Table 3) integrates agent triggers, governance filters, model selection, feedback, and cross-agent escalation, ensuring that advisories, rerouting, or hazard protocols activate only when thresholds are crossed, with stability maintained via hysteresis and cooldown controls.

5.3. Agentic AI pipeline with use case context

In order to give the formal definition of the smart-city disaster management ecosystem, this section introduces the mathematical definition of the agentic AI pipeline implemented in the usecase agents of Weather, Traffic, and Safety (Fire/Smoke). The pipeline is hybrid, integrating trained and pre-trained models of ML and Computer vision to do perception and domain-specific prediction (e.g., XGBoost regime classification and YOLO-based detection), and pre-trained Large Language Models (LLMs) to do contextual reasoning, policy interpretation, and governance-sensitive decision support. At the time of every decision, a deterministic (hard-coded) prompt template is filled with real-time API values, verified ML outputs and a similar structured prompt is released to three LLMs GPT, Grok, and DeepSeek to receive independent risk trust ratings and explanations. The resultant multi-LLM outputs are then compared to reference governance values, allowing the MAE-based selection and alignment feedback-based on the result without a parameter change of LLM. This rigid division of perception/prediction (ML) and contextual reasoning (LLMs) also facilitates explainability, reproducibility, and regulation-consistent behaviour in heterogeneous smart-city settings.

Table 2
Overview of domain agents and governance authority with data scope, outputs, performance, and ledger anchoring.

Agent / Authority	Data Scope	Primary Outputs	Performance and Ledger
Weather Agent	Open-Meteo (2014–2025): temperature, precipitation, humidity, wind, cloud cover, UV	Classification (Normal, Rain, Heavy Rain, Heatwave); trust–risk rationale	XGBoost (Accuracy: train = 0.998, test = 0.978); Agentic Chain
Traffic Agent	Vehicle counts: CCTV streams; congestion threshold: 15 vehicles/100 m	Congestion, risk and trust; routing recommendations	Pre-trained YOLOv8 (mAP@50 = 0.975, mAP@50–95 = 0.742); Agentic Chain
Safety Agent	Fire and smoke detections: CCTV streams	Hazard extent with confidence-based annotations; risk–trust	Pre-trained YOLO11 "Flare Guard" (mAP@50 = 0.770, mAP@50–95 = 0.492); Agentic Chain
SORA Governance	Cross-agent oversight and policy enforcement	MAE-based LLM selection, corrective feedback, and global directives	Governance records; *SORA Chain*

Table 3
SORA unified policy matrix: agent triggers, governance filters, and escalation (using R, T, and ecosystem metrics from Defs. 4–7).

Scope	ID	Trigger / Rule	Primary Action	Notes
Agent-Level Policies				
Weather	W1	$R^{\text{Wea}}(t) > 0.60 \wedge T^{\text{Wea}}_{\text{Overall}}(t) < 0.65$	Flood/heatwave advisories	Select LLM closest to SORA reference (post S1)
Traffic	T1	$R^{\text{Tra}}(t) \geq 0.95 \wedge T^{\text{Tra}}_{\text{Overall}}(t) < 0.65$	Rerouting, signal tuning	Select closest-to-SORA LLM (post S1)
Fire	F1	$R^{\text{Fire}}(t) \geq 0.95 \wedge T^{\text{Fire}}_{\text{Overall}}(t) > 0.65$	Dispatch, evacuation	Enforce most consistent LLM (post S1)
Governance-Level Policies				
SORA	S1	Risk-Trust Gate: admit if $\lvert\Delta R\rvert \leq 0.07$, $\lvert\Delta T\rvert \leq 0.05$, and $T_i^{\text{Overall},j} \geq \tau_T(i)$ (Wea:0.60, Tra:0.55, Fire:0.65)	Admit candidates	Code vars: `risk_thresh` = 0.07, `trust_thresh` = 0.05
SORA	S2	Selection & Tie-break: choose $\max T$; if tie, pick $\min\lvert\Delta R\rvert$ (ties within 0.01 equal). Fallback (Fire only): nearest $T \geq \tau_T$, else nearest T[a]	Select best model	No fallback for Weather/Traffic
SORA	S3	Error-Directed Feedback: $\Delta R = R_{\text{SORA}} - R_j$, $\Delta T = T_{\text{SORA}} - T_j$	Feedback to non-selected	Apply 50% adjustment; clip $[0, 1]$
Cross-Agent / Ecosystem Policies				
Cross-Agent	S4	≥ 2 agents with $R^i(t) > 0.80$	Joint actuation (reroutes, co-alerts)	T^i_{Overall} modulates confidence
Ecosystem	S5	$R_{\text{Ecosystem}}(t) > 0.70 \wedge T_{\text{Ecosystem}}(t) \geq 0.60$	City-wide escalation	Human confirmation if $T_{\text{Ecosystem}} < 0.60$
Safety	S6	Hysteresis $h = 0.05$, cooldown $\Delta t_{\min} = 15$ min	Stability / anti-chatter	Prevent oscillations and alert spam

[a] *Abbrev.:* Wea = Weather, Tra = Traffic, Fire = Fire(Safety). R = Overall Risk (Def. 4), T = Overall Trust (Def. 6).

5.3.1. Dataset and predictive model formulation

Let t denote a discrete decision interval in the smart-city operational timeline. Each domain agent $A_i \in \{\text{Weather}, \text{Traffic}, \text{Safety}\}$ acquires real-time observations from heterogeneous data sources, including external APIs and physical sensors. This ecosystem are represented as:

$$\mathcal{D}_i(t) = \{x_1(t), x_2(t), \ldots, x_m(t)\}, \tag{12}$$

where $\mathcal{D}_i(t)$ consists of structured numerical signals (e.g., temperature, precipitation, vehicle counts), semi-structured metadata (e.g., timestamps, geolocation), and unstructured inputs (e.g., traffic or CCTV frames). To ensure deterministic, explainable, and domain-accurate perception, each agent employs a trained or pre-trained ML/CV model $g_i(\cdot)$ that maps raw observations to domain-specific predictions:

$$y_i(t) = g_i(\mathcal{D}_i(t)), \tag{13}$$

where $y_i(t)$ denotes the validated predictive output consumed by downstream reasoning components.

Weather dataset and XGBoost model. For the Weather agent, $\mathcal{D}_{\text{Weather}}(t)$ consists of meteorological variables obtained from the *Open-Meteo API*, a publicly available and widely used weather data service. At each decision interval t, the weather observation vector is represented as:

$$\mathcal{D}_{\text{Weather}}(t) = [T(t),\ P(t),\ H(t),\ W(t),\ UV(t)], \tag{14}$$

where $T(t)$ denotes ambient temperature (°C), $P(t)$ is precipitation (mm/day), $H(t)$ is relative humidity (%), $W(t)$ is wind speed (m/s), and $UV(t)$ is the ultraviolet index.

For example, a sampled observation at time t may be expressed as:

$$\mathcal{D}_{\text{Weather}}(t) = [41.2,\ 48.6,\ 72,\ 6.3,\ 9.1], \tag{15}$$

indicating extreme heat with heavy precipitation and high UV exposure. These continuous-valued features are mapped to discrete weather regimes (e.g., *Normal*, *Rain*, *Heavy Rain*, *Heatwave*) using a trained XGBoost classifier:

$$y_{\text{Weather}}(t) = g_{\text{XGB}}\big(\mathcal{D}_{\text{Weather}}(t)\big), \tag{16}$$

where $y_{\text{Weather}}(t)$ denotes the predicted regime label. XGBoost is selected for its ability to capture non-linear interactions between meteorological variables, robustness to noisy environmental data, and strong performance on structured, tabular data-sets, properties that are essential for reliable perception in governance-sensitive smart-city applications.

Traffic dataset and YOLO-based detection model. For the Traffic agent, $\mathcal{D}_{\text{Traffic}}(t)$ consists of visual data streams acquired from roadside surveillance cameras and urban traffic monitoring systems deployed across the city. These data sources provide real-time image or video frames representing vehicular flow conditions at time t. Formally, the traffic observation at time t is expressed as:

$$\mathcal{D}_{\text{Traffic}}(t) = \{I_k(t) \mid k = 1, 2, \ldots, N\}, \tag{17}$$

where $I_k(t)$ denotes the k-th image frame captured within a fixed aggregation window (e.g., 100-meter road segment).Each frame is processed using a pre-trained YOLOv8 object detection model to identify and localize vehicles in real time:

$$y_{\text{Traffic}}(t) = g_{\text{YOLO}}\big(\mathcal{D}_{\text{Traffic}}(t)\big), \tag{18}$$

where $y_{\text{Traffic}}(t)$ represents the estimated vehicle count and associated confidence scores.

The YOLO detector outputs a set of bounding boxes:

$$y_{\text{Traffic}}(t) = \{(b_j, c_j) \mid j = 1, 2, \ldots, M\}, \tag{19}$$

where $b_j = (x_j, y_j, w_j, h_j)$ denotes the bounding box coordinates of the j-th detected vehicle and $c_j \in [0, 1]$ is the corresponding detection confidence.

The aggregated vehicle density for a road segment is computed as:

$$V(t) = \sum_{j=1}^{M} \mathbb{I}(c_j \geq \theta_c), \tag{20}$$

where $\mathbb{I}(\cdot)$ is the indicator function and θ_c is the confidence threshold (set to 0.5 in our implementation).

For example, a sampled traffic observation may yield:

$$y_{\text{Traffic}}(t) = \{(b_1, 0.91), (b_2, 0.88), \ldots, (b_{20}, 0.84)\}, \tag{21}$$

resulting in an estimated vehicle count of $V(t) = 20$ vehicles per 100-meter segment.YOLOv8 is selected for traffic perception due to its single-stage detection architecture, which enables low-latency inference and high detection accuracy in dense urban environments. Its ability to jointly localize and classify multiple vehicles within a single forward pass makes it particularly suitable for real-time congestion estimation and agentic decision pipelines, where timely and consistent perception is critical for governance-aware traffic management.

Safety dataset and fire/smoke detection model. For the Safety agent, $\mathcal{D}_{\text{Safety}}(t)$ consists of visual data streams obtained from urban CCTV cameras deployed in public spaces, commercial districts, and critical infrastructure. These cameras provide continuous image or video frames that capture potential hazardous events such as fire or smoke. Formally, the safety observation at time t is represented as:

$$\mathcal{D}_{\text{Safety}}(t) = \{F_k(t) \mid k = 1, 2, \ldots, N\}, \tag{22}$$

where $F_k(t)$ denotes the k-th video frame captured within a short temporal window.

Each frame is processed using a pre-trained YOLO-based fire and smoke detection model to identify hazardous regions in real time:

$$y_{\text{Safety}}(t) = g_{\text{Fire}}\big(\mathcal{D}_{\text{Safety}}(t)\big), \tag{23}$$

where $y_{\text{Safety}}(t)$ denotes the detected hazard instances and their associated confidence scores.

The detector produces a set of hazard predictions:

$$y_{\text{Safety}}(t) = \{(b_j, c_j, \ell_j) \mid j = 1, 2, \ldots, M\}, \tag{24}$$

where $b_j = (x_j, y_j, w_j, h_j)$ represents the bounding box coordinates of the j-th detected region, $c_j \in [0, 1]$ is the detection confidence, and $\ell_j \in \{\text{fire}, \text{smoke}\}$ denotes the hazard class label.A hazard event is confirmed when the maximum detection confidence exceeds a predefined threshold:

$$H(t) = \begin{cases} 1, & \text{if } \max_j c_j \geq \theta_h, \\ 0, & \text{otherwise}, \end{cases} \tag{25}$$

where θ_h is the hazard confidence threshold (set to 0.5 in our implementation).For example, a sampled safety observation may yield:

$$y_{\text{Safety}}(t) = \{(b_1, 0.74, \text{smoke}), (b_2, 0.55, \text{fire})\}, \tag{26}$$

indicating simultaneous smoke and fire detections within the monitored scene.

YOLO-based architectures are selected for fire and smoke detection due to their ability to perform real-time object localization and classification in visually complex environments. Their single-stage design enables rapid hazard identification with bounded latency, which is essential for safety-critical applications such as emergency response and evacuation coordination.

5.3.2. Multi-LLM reasoning layer: GPT-4, Grok, and DeepSeek

While trained and pre-trained ML/CV models provide deterministic perception and prediction, contextual interpretation, policy reasoning, and trust–risk synthesis require higher-level semantic intelligence. To address this, the proposed ecosystem employs three pre-trained Large Language Models (LLMs) ; GPT-4, Grok, and DeepSeek as parallel agentic reasoning engines. These models are not used for prediction and are not fine-tuned at the parameter level. Instead, they operate exclusively as contextual reasoning functions over validated ML/CV outputs and governance constraints.

Mathematical abstraction of LLM reasoning. At each decision interval t, a domain agent A_i constructs a structured prompt $P_i(t)$ and submits it concurrently to all three LLMs. Each LLM $j \in \{\text{GPT-4}, \text{Grok}, \text{DeepSeek}\}$ is modeled as a semantic reasoning function:

$$f_j : P_i(t) \rightarrow \langle R_{i,j}(t),\ T_{i,j}(t),\ E_{i,j}(t) \rangle, \tag{27}$$

where $R_{i,j}(t) \in [0, 1]$ denotes the inferred operational or environmental risk, $T_{i,j}(t) \in [0, 1]$ denotes the inferred trust or confidence level, and $E_{i,j}(t)$ is a natural-language explanation constrained to the provided evidence.

By evaluating multiple LLMs in parallel, the system captures semantic diversity arising from differences in training corpora, reasoning priors, and internal representations, which is critical for mitigating hallucination and model-specific bias in safety-critical smart-city operations.

Hard-coded prompt structure with variable injection. To ensure determinism, reproducibility, and governance alignment, all LLM interactions follow a hard-coded prompt template $\Phi(\cdot)$ with strictly defined fields. Dynamic values are injected only through validated API data and ML/CV model outputs. Formally, the prompt is constructed as:

$$P_i(t) = \Phi\big(\mathcal{D}_i(t),\ y_i(t),\ \Pi_i,\ \Theta_i\big), \tag{28}$$

where $\mathcal{D}_i(t)$ represents raw observations, $y_i(t)$ denotes ML predictions (e.g., weather regimes, vehicle counts, hazard detections), Π_i encodes domain and regulatory policies, and Θ_i specifies trust–risk thresholds and output constraints.

Prompt template (Hard-coded portion). The hard-coded portion enforces strict reasoning boundaries and output formats:

> **SYSTEM ROLE:** You are an agentic AI operating within a smart-city governance framework. You must reason only over the provided inputs. Do not assume or hallucinate missing information. **TASK:**
>
> - Estimate Risk by using given formulae $R \in [0, 1]$
> - Estimate Trust by using given formulae $T \in [0, 1]$
> - Provide a concise justification grounded in the inputs
>
> **OUTPUT FORMAT (STRICT JSON):**

```
{
  ''R'': <float>,
  ''T'': <float>,
  ''explanation'': ''<text>''
}
```

This structure ensures that all LLM outputs are machine-verifiable and directly comparable across models.

Variable portion (Runtime injection). At runtime, real-time values replace placeholders in the prompt, for example:

$$\begin{aligned} \mathcal{D}_{\text{Traffic}}(t) &= \{\text{camera frames},\ \text{timestamp}\}, \\ y_{\text{Traffic}}(t) &= \text{vehicle count} = 20, \\ \Theta_{\text{Traffic}} &= \{\theta_R = 0.85,\ \tau_T = 0.55\}. \end{aligned} \tag{29}$$

These injected values ensure that LLM reasoning is grounded in real sensor evidence rather than free-form inference.

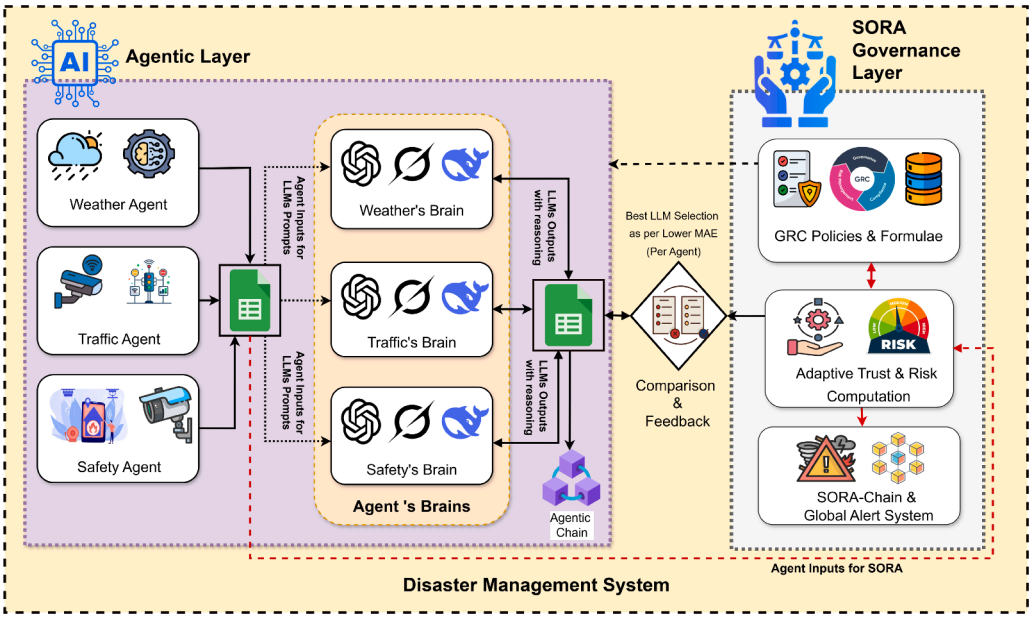

Fig. 7. End-to-end workflow of the SORA-ATMAS architecture showing API-based orchestration of LLMs, local agentic validation, and global SORA enforcement with dual blockchain anchoring. The figure demonstrates how agent-level decisions are first validated locally through LLM-based reasoning and then subjected to global trust–risk evaluation and policy enforcement, enabling auditable and coordinated governance across heterogeneous disaster-management agents.

Governance-guided model selection. The governance agent (SORA) computes reference trust–risk values $\langle R_{\text{ref}}(t), T_{\text{ref}}(t)\rangle$ and evaluates each LLM output using Mean Absolute Error:

$$\text{MAE}_j(t) = \frac{1}{2}\left(|R_{i,j}(t) - R_{\text{ref}}(t)| + |T_{i,j}(t) - T_{\text{ref}}(t)|\right). \tag{30}$$

The selected model is:

$$j^*(t) = \arg\min_j \text{MAE}_j(t), \tag{31}$$

subject to trust and risk gating constraints. Non-selected models receive error-directed feedback, which is injected into subsequent prompts, enabling iterative semantic alignment without modifying LLM parameters.

Rationale for multi-LLM design. GPT-4 provides stable, policy-aware reasoning suitable for baseline governance alignment. Grok introduces diversity and sensitivity to dynamic conditions, which is beneficial in volatile mobility scenarios. DeepSeek complements the ensemble with strong consistency in hazard-centric narratives, particularly in Weather and Safety domains. The coordinated use of these models, governed through quantitative validation and feedback, enables robust, explainable, and regulation-aligned agentic decision-making across heterogeneous smart-city services.

6. Workflow explanation and execution

The workflow integrates decentralized AI agents, LLM-based reasoning, and centralized governance to operationalize adaptive trust management in disaster response. As shown in Fig. 7, local agents process sensor data, invoke multiple LLMs for semantic reasoning, and generate structured risk-trust outputs that are relayed to the SORA Governance Layer for validation, compliance enforcement, and escalation. Unlike static pipelines, this workflow supports context-aware, risk-sensitive decisions across both stable and high-risk scenarios. The closed feedback loop $(\Delta R, \Delta T)$ iteratively returned to agent nodes drives convergence and MAE reduction (Section 7). The setup was implemented on a distributed testbed (as shown in Table 6).

Agent-level workflow (Local processing on PC-A).

1. Each domain-specific agent (Weather, Traffic, Safety) collects contextual sensor data (e.g., temperature, congestion levels, fire/smoke frames) and records it into structured logs on Google Sheets. A sample message structure is defined as:

$$M = \{\text{AID}, D, t, \lambda\}, \tag{32}$$

where AID denotes the agent identifier, D represents the sensor data payload, t is the timestamp, and λ encodes local compliance metadata.
2. The logged entries are transmitted via API calls to multiple LLMs (ChatGPT, Grok, DeepSeek), which act as the semantic reasoning "brains." Each model computes domain-specific trust and risk components:

$$\{R^i_{\text{Env}},\ R^i_{\text{Service}},\ T^i_{\text{Ctx}},\ T^i_{\text{HRT}}\}, \tag{33}$$

which are then aggregated into overall trust-risk scores $R^i(t)$ and $T^i(t)$ according to the definitions in Section 4.4.
3. The agent signs and forwards the per-LLM outputs $\{R^i(t), T^i(t)\}$ to SORA. The signed packet is anchored on the Agentic Blockchain for provenance.

SORA governance workflow (on PC-B).

1. At the governance node, SORA first ingests raw sensor values transmitted by the agents. These inputs are checked using the policies given by the Security Policy Engine for authentication, authorization, and access control and by the Cross-Domain Operational Policy Engine for domain-specific compliance. Only packets that pass both checks are admitted, and from these, SORA computes the governance-level reference risk R^{ref} and trust T^{ref} values according to the definitions in Section 4.4, which form the baseline for comparison.
2. SORA then receives the signed per-LLM outputs $\{R^i(t), T^i(t)\}$ from each agent. These packets are again verified for provenance, authorization, and compliance. Invalid or non-compliant packets are discarded before further processing.
3. For each valid packet, SORA calculates the Mean Absolute Error (MAE) between an LLM's reported outputs and its governance ref-

erence values:

$$\mathrm{MAE}_j = \frac{1}{2}\Big(|R_j - R_{\mathrm{ref}}| + |T_j - T_{\mathrm{ref}}|\Big), \tag{34}$$

where j indexes the LLMs. These MAE scores are used in policy step S2 (model selection) and step S3 (error-directed feedback).

4. The Adaptive Trust and Risk Enforcement Engine applies global thresholds as specified in the policy matrix. Formally:

$$\texttt{Decision} = \begin{cases} \texttt{deny}, & T^i < \theta_T, \\ \texttt{restrict}, & R^i > \theta_R, \\ \texttt{approve}, & \text{otherwise.} \end{cases} \tag{35}$$

This ensures that high-risk or low-trust outputs cannot bypass systemic safeguards.

5. Cross-agent and ecosystem-level policies are then enforced. If at least two agents report high risk ($R^i > 0.80$), joint actuation is triggered; if the ecosystem risk exceeds 0.70 with ecosystem trust ≥ 0.60, global level escalation follows (with human confirmation required if trust is lower).
6. All governance decisions, including model selection, feedback, joint actions and escalation outcomes are immutably anchored on the SORA Blockchain. In addition, all dispatched alerts (such as flood advisories, traffic reroutes and fire evacuations) are also logged on-chain for provenance and audit while being simultaneously delivered to stakeholders via API channels.

This layered synchronization between PC-A and PC-B is maintained through structured data exchange (Google Sheets API), tamper-resistant anchoring on the Agentic and SORA Blockchains, and lightweight messaging for the timely dissemination of alerts and directives.

7. Results and discussion

The evaluation of the SORA-ATMAS framework is conducted in two dimensions: *Semantic* alignment of local LLMs with governance-enforced reference outputs and *Operational performance* of SORA as a centralized authority within a distributed multi-agent setting including cross-domain interoperability for coordinated escalation. On the semantic side, the focus lies on how candidate models progressively converge toward policy-aligned outputs under governance supervision. On the operational side, we analyze throughput, execution time, and governance-induced delay across varying workloads to capture the trade-offs between oversight, responsiveness and blockchain anchoring. Together, these results demonstrate SORA's ability to balance correctness, safety, and scalability in smart-city disaster governance.

7.1. Agent-level evaluation of LLMs

Three domain agents (Weather, Traffic, and Fire/Smoke) were evaluated across three iterations (Iter. 0-2). Each agent processed 20 structured requests per iteration, yielding 60 total observations per model per iteration. This smaller, controlled batch size was used for semantic convergence analysis rather than large-scale throughput testing. The selection of candidate LLMs (GPT, Grok, DeepSeek) was governed by two joint criteria:

1. Minimizing the Mean Absolute Error (MAE) with respect to SORA's reference values, and
2. Satisfying governance policies (S1-S3) as given in Table 3.

Specifically, the policies enforced a $\texttt{risk\_threshold} = 0.07$, a $\texttt{trust\_threshold} = 0.05$, and domain-specific trust baselines $\tau_t = \{0.60, 0.55, 0.65\}$ for Weather, Traffic, and Fire/Smoke, respectively. When no candidate satisfied S1, the fallback mechanism was activated, allowing a model to be selected solely for meeting the trust baseline τ_t, even if its MAE was not minimal. Error-directed feedback $(\Delta R, \Delta T)$ was iteratively applied with an adjustment factor of 0.5 (clipped to $[0, 1]$), driving non-selected models toward convergence. Figs. 8 and 9 show the trajectory of risk-trust values and the evolution of MAE distributions.

Weather agent. The Weather agent maintained low R_{Overall} values across iterations. At Iter. 0, GPT achieved the lowest MAE (risk and trust). By Iter. 1, error feedback allowed DeepSeek to significantly reduce its trust MAE, and by Iter. 2, DeepSeek surpassed GPT with the lowest trust MAE while maintaining stable risk alignment. Risk MAE remained negligible, confirming stable conditions.

Traffic agent. Traffic data showed a high-risk plateau ($R \approx 0.85$). Grok consistently achieved the lowest MAE across iterations, dominating in Iter. 0 and Iter. 1. GPT and DeepSeek reduced their trust MAE over iterations, as seen in Figs. 8 and 9. By Iter. 2, Grok retained dominance, illustrating its stability under congestion. Risk MAE remained near zero, making trust MAE the decisive factor.

Safety (Fire/Smoke) agent. The Fire/Smoke agent transitioned from high-risk/low-trust $(R > 0.9, T < 0.2)$ to low-risk/high-trust $(R \approx 0.1, T > 0.6)$. At Iter. 0, most models failed the S1 trust gate $(\tau_T = 0.65)$, activating the fallback policy, which selected DeepSeek despite its higher MAE. By Iter. 1, DeepSeek maintained dominance while lowering its trust MAE. By Iter. 2, more models satisfied S1, but DeepSeek still outperformed with the lowest trust MAE.

Key trends. Across all agents, SORA governance mechanisms (S1 gating, S2 tie-breaking, S3 feedback, fallback) consistently reduced trust mean absolute error (MAE) while maintaining stable risk MAE. These findings indicate that governance ensured threshold compliance and minimized MAE, guiding models toward SORA reference behavior and maintaining safety in dynamic conditions. Semantic alignment at the agent level enables coherent cross-domain reasoning, allowing SORA to integrate independent risk and trust assessments.

Governance-enforced operational convergence. The sequential decline in mean absolute error (MAE) observed in Figs. 8 and 9 arises from the policy-guided feedback mechanism defined in Algorithm 2(steps S2–S3) and summarized in Table 3. Specifically, the 50% adjustment factor applied to the system error signals $(\Delta R, \Delta T)$ systematically steers non-selected large language models (LLMs) toward the SORA reference model. The fallback mechanism, which is triggered when no model satisfies policy S1, ensures continuous system operation. More importantly, operational safety is preserved even in the absence of complete semantic alignment, as only SORA-approved outputs are enacted and are immutably logged on-chain. Consequently, SORA-ATMAS demonstrates convergence through empirically observed, governance-enforced behavior rather than asymptotically guaranteed theoretical convergence, while maintaining bounded and secure behavior at every iteration.

MAE reductions and statistical analysis

The MAE reductions across iterations as shown in Figs. 8, and 9 were analyzed using boxplots and Wilcoxon Signed-Rank tests. The boxplots Fig. 10 show the variance in MAE Total reductions across iterations for the Weather, Traffic, and Fire/Smoke agents. These reductions are compared against the *single-LLM baseline (GPT)* and the *oracle-selection upper bound* to assess model performance.

The Wilcoxon Signed-Rank test results for MAE reductions across iterations are presented in Table 4. P-values below 0.05 indicate statistically significant reductions in MAE. Except for GPT in certain agent iterations (e.g., Weather Agent), all models show significant reductions.

Additionally, the Mean MAE for the single-LLM baseline (GPT) and the oracle-selection upper bound are provided in Table 5. The single-LLM baseline (GPT) is used as a comparison to assess model performance, and the oracle-selection upper bound represents the best possible model selection.

Combining the graphical evidence (Figs. 8 and 9), statistical tests (Table 4) and policy-constrained selection (Algorithm 2, Table 3), prove that SORA-ATMAS satisfies the operational convergence of governance without violating safety invariants on each step.

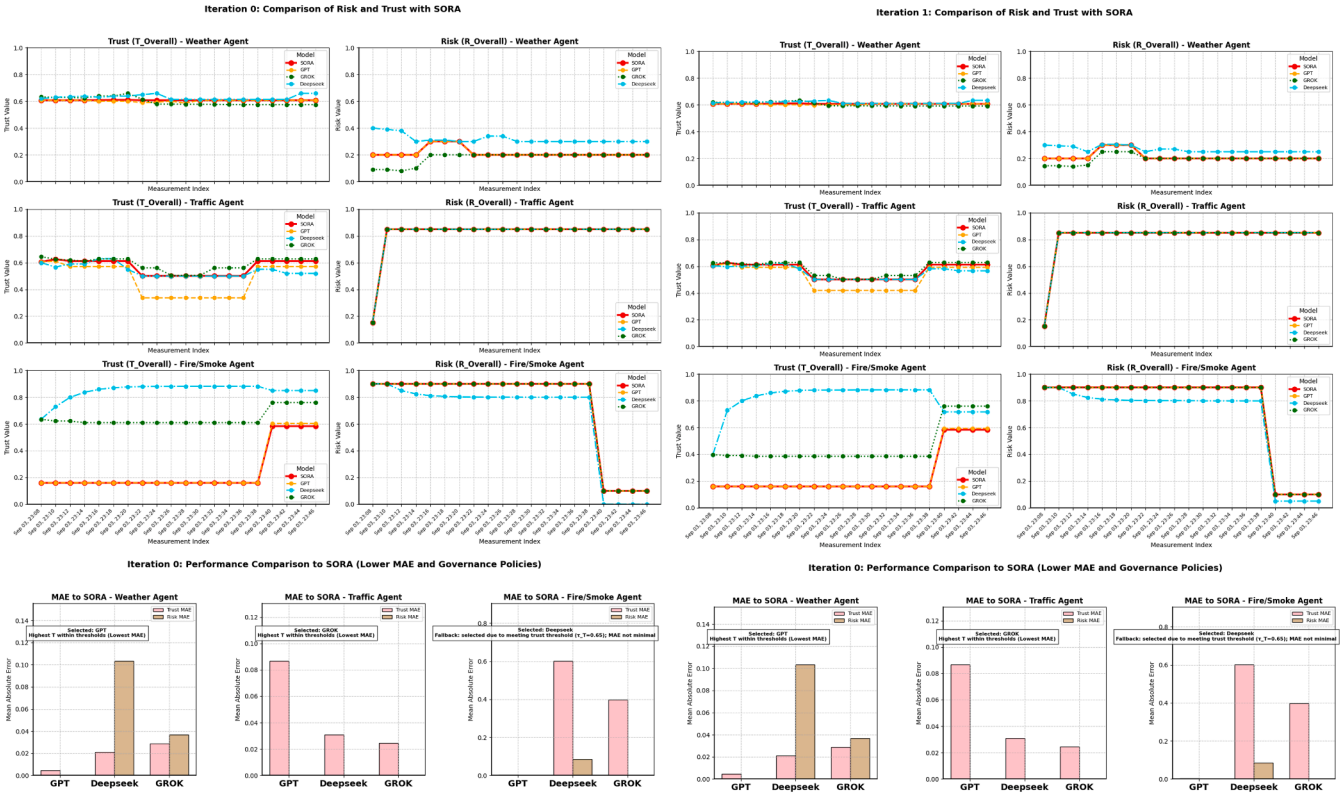

Fig. 8. Iterations 0 and 1: Comparison of candidate LLMs with SORA across Weather, Traffic, and Fire/Smoke agents. Top: Trust and Risk trajectories. Bottom: MAE-to-SORA distributions. The results show that SORA-guided selection and feedback rapidly reduce trust and risk divergence, providing early alignment of candidate LLMs toward governance-consistent behavior.

Table 4
Wilcoxon signed-rank test results showing the decrease in MAE across iterations. Statistically significant improvements at low p-values (< 0.05) confirm the convergence trends in Figs. 8–9 and the effectiveness of the governance-based feedback loop (Algorithm 2, S2–S3).

Agent	Model	P-Values
Weather	GPT	0.1088 / 0.1797
Weather	DeepSeek	0.000078 / 0.000120
Weather	GROK	0.000124 / 0.000124
Traffic	GPT	0.000087 / 0.000087
Traffic	DeepSeek	0.000077 / 0.000166
Traffic	GROK	0.006459 / 0.002968
Fire/Smoke	GPT	0.0455 / 0.0455
Fire/Smoke	DeepSeek	0.0339 / 0.0339
Fire/Smoke	GROK	0.000175 / 0.000175

Table 5
Mean MAE for Single-LLM Baseline and Oracle-Selection Upper Bound.

Agent	Mean MAE
Weather	0.00445 (GPT), 0.0039 (Oracle-Selection)
Traffic	0.08685 (GPT), 0.0072 (Oracle-Selection)
Fire/Smoke	0.00400 (GPT), 0.0040 (Oracle-Selection)

7.2. Performance evaluation of SORA

After establishing agent-level semantic alignment, we evaluate SORA's runtime performance as a governance authority. Beyond local processing, SORA validates reports, enforces policies, and anchors decisions on the blockchain, introducing governance overhead measured by three key metrics:

1. **Throughput (T):** The effective rate of governance-approved requests per second:
$$T = \frac{N}{t^{\text{last decision}} - t^{\text{first request}}}, \tag{36}$$
where N is the number of processed requests.
2. **Execution time (ET):** The average per-request latency at the agent level, comprising policy fetching, local risk-trust computation, and secure logging to the Agent-Chain:
$$\text{ET} = t^{\text{fetch}} + t^{\text{compute}} + t^{\text{Agentic-Chain log}}. \tag{37}$$
3. **Operational delay (D):** The additional per-request overhead introduced by SORA governance, consisting of global validation, MAE-based LLM selection, corrective feedback dissemination, final decision synthesis, and SORA-Chain anchoring:
$$D = t^{\text{validate}} + t^{\text{MAE-select}} + t^{\text{feedback}} + t^{\text{final-decision}} + t^{\text{SORA-Chain log}}. \tag{38}$$

Experimental testbed configuration. The performance evaluation was conducted on a distributed two-layer testbed consisting of multiple agent nodes and a single governance node. The hardware, network, and blockchain configurations used for experiments are summarized in Table 6.

Experimental setup. The performance evaluation used synthetic workloads simulating structured observations from domain agents such as Weather, Traffic, and Safety, with request sizes of 100, 500, 1000, and 2000. To assess scalability, a fixed workload of 100 requests was used to measure throughput (T), execution time (ET), and governance delay (D) across different agent configurations. Larger configurations with six and nine agents were projected using an analytical M/M/c queueing model [54] calibrated to the measured three-agent results, as hardware

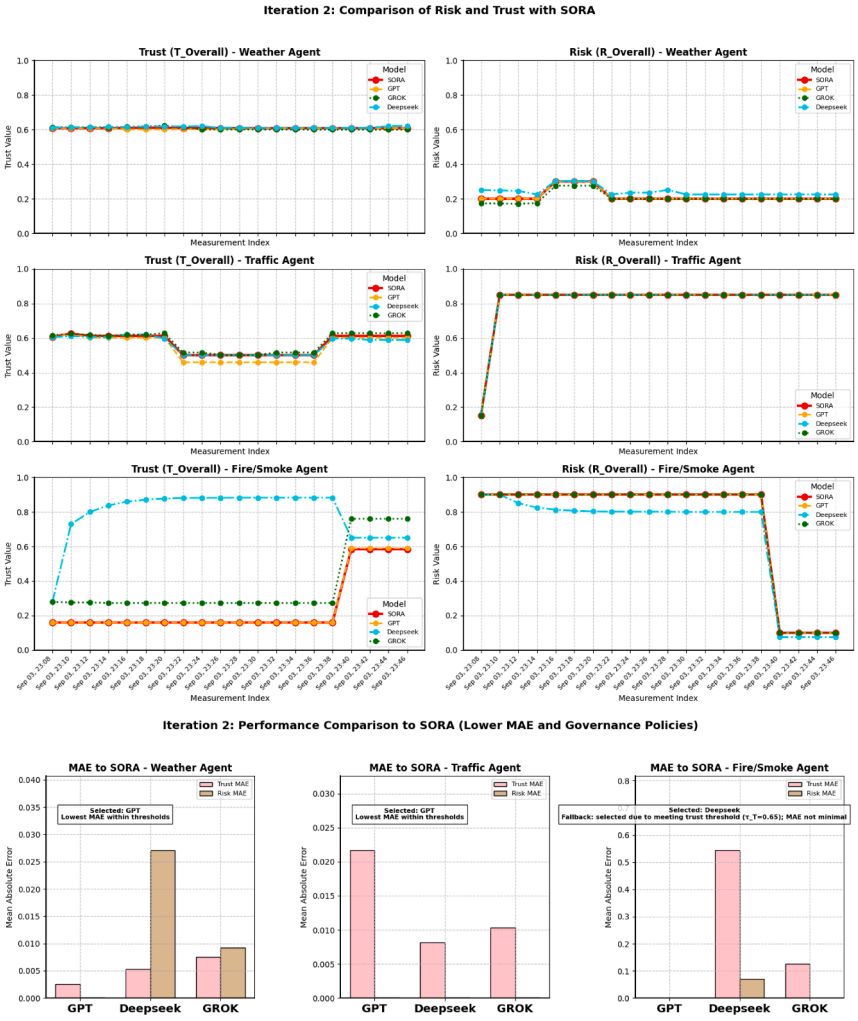

Fig. 9. Iteration 2: Convergence of candidate LLMs toward SORA with stable risk alignment and further reduction in trust MAE. This figure demonstrates convergence of the candidate models under repeated governance feedback, indicating stabilization of both trust and risk signals across all agent domains.

Table 6
Hardware configuration for SORA evaluation.

Component	Specifications
Agent Nodes (3× PC-A)	Intel® i7-12700K (3.8 GHz, 32 threads); 32 GB RAM; 1 TB SSD; Ubuntu 22.04.
Governance Node (PC-B)	AMD Ryzen 9 5900X (64 threads); 64 GB RAM; 2 TB SSD; Ubuntu 22.04.
Network	Gigabit LAN, <2 ms latency.
Blockchain	Multichain: synchronous (SORA-Chain), asynchronous/batched (Agentic-Chain).

Table 7
Runtime performance of SORA including blockchain anchoring and decision synthesis.

Requests	T (req/s)[a]	ET / req (ms)[a]	D / req (ms)[a]
100	17.2	58	21
500	16.3	61	32
1000	15.2	65	52
2000	13.8	72	92

[a] **Notes:** T = Throughput, ET = Execution Time per request, and D = Operational/Governance Delay per request.

constraints limited physical scalability beyond three nodes. Timers at both the agent and governance layers recorded ET and D, while T was derived from the total wall-clock duration.

Runtime performance and scalability of SORA. The run-time performance of SORA, including blockchain-anchored decision-making and decision synthesis, was assessed with a series of synthetic workloads of various sizes. The results for throughput, execution time (ET), and governance delay (D) of the 3-agent configuration are provided in Table 7.

The scalability of SORA was further analyzed across different agent configurations (3, 6, and 9 agents). For the 3-agent configuration, we measured throughput, execution time, and governance delay. For the 6- and 9-agent configurations, the results were projected using an analytical M/M/c queueing model, as shown in Table 8.

Throughput trends. As shown in Table 7 and Fig. 11, throughput declines moderately with workload size, from 17.2 req/s at 100 requests to 13.8 req/s at 2000, primarily due to serial governance operations and dual-chain anchoring. Throughput remains above 13 req/s even under the maximum load, confirming practical scalability for near-real-time applications.

Execution time analysis. Execution time at the agent level increases modestly (58 ms to 72 ms) with higher workloads, indicating that policy

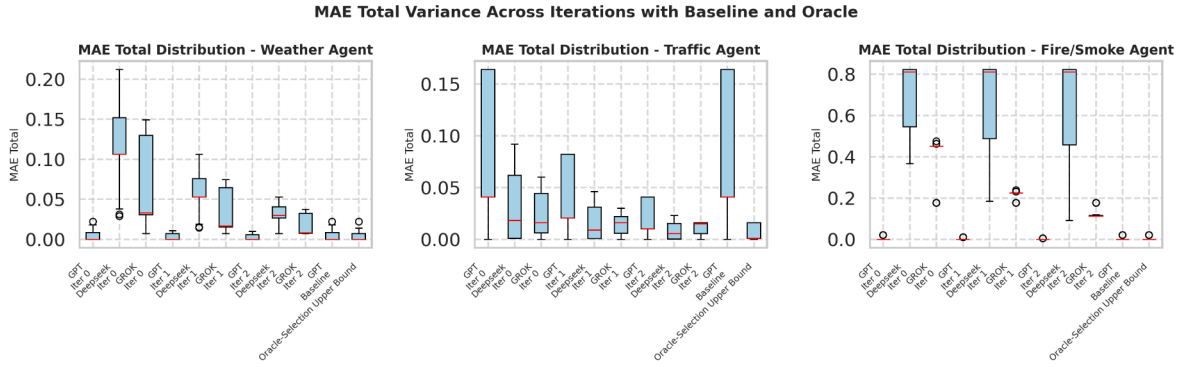

Fig. 10. MAE distribution across iterations for Weather, Traffic, and Fire/Smoke agents, with baseline (GPT) and oracle upper bound for comparison. The progressive reduction in MAE across iterations confirms that SORA-guided governance systematically improves alignment accuracy while maintaining bounded variance relative to oracle behavior.

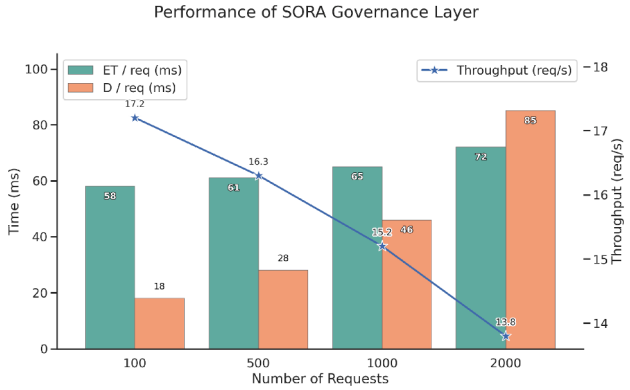

Fig. 11. Performance of the SORA governance layer for the three-agent deployment, showing execution time (ET), governance delay, and throughput. It demonstrates that SORA maintains predictable latency and stable throughput as request volume increases, validating its suitability for real-time governance and escalation in smart-city disaster scenarios.

Table 8
Scalability analysis across different agent configurations. Projections are simulation-based on fixed 100 requests.

Metric	3 Agents (Measured)	6 Agents (Projected)	9 Agents (Projected)
T (req/s)[a]	17.2	15.8	14.1
ET / req (ms)[a]	58	64	71
D / req (ms)[a]	21	38	62

[a] **Notes:** T = Throughput, ET = Execution Time per request, and D = Operational/Governance Delay per request.

fetching, local risk-trust computation, and Agentic-Chain logging scale efficiently on CPU-only agent nodes. This confirms the lightweight nature of local agent pipelines and efficient blockchain integration.

Operational delay trends. Governance/Operational delay (D) rises more sharply (21 ms to 92 ms) as workloads grow, reflecting the additional costs of validation, model selection, and synchronous SORA-Chain anchoring. Despite this growth, delay remains under 100 ms for the tested range, supporting the feasibility of responsive governance at modest agent scales.

Scalability discussion. Analytical performance data presented in Table 8 extend the measured performance results to larger deployment sizes. Under comparable agent loads and networking conditions, SORA is expected to support more than 14 requests per second for deployments with nine agents. The increased synchronous anchoring and coordination overhead associated with governance leads to higher governance latency. The identified scalability limitations of the governance layer in large-scale deployments will be addressed through proposed architectural enhancements, including SORA sharding, partial asynchronicity, and multichain partitioning.

In general, SORA finds an effective middle ground to ensure both runtime responsiveness and regulatory monitoring; the distributed testbed (Table 6) shows that CPU-only agent nodes can provide a high-quality execution and verifiable blockchain anchoring in a resource-efficient manner; the empirical data collected from 3-agents as well as the simulations demonstrate a conservative, but practical, view of SORA's scalability and identify paths toward optimizing it for larger-scale smart-city deployments

7.3. Baseline performance validation with GPT-only governance

To validate the realism of the reported throughput and governance delay in a dual-chain, multi-agent setting, we perform a comparative runtime analysis against a recent state-of-the-art blockchain-based smart-city system that integrates GPT models for smart contract optimization [55]. Unlike the full SORA framework, which supports MAE-based multi-LLM selection, this analysis evaluates SORA in a *GPT-4-only configuration*, while keeping all governance logic, policy enforcement, and dual-chain anchoring unchanged. This provides a controlled baseline comparison focused on governance overhead and end-to-end decision latency.

Comparison baseline and experimental alignment. The reference system in [55] evaluates GPT-2, GPT-3, and GPT-4 optimized smart contracts on a permissioned MultiChain blockchain for secure smart-city service interoperability. Both systems share key architectural characteristics, including permissioned blockchain execution, API-driven AI integration, and governance-oriented transaction flows. However, the benchmark system primarily optimizes cryptographic message exchange and contract execution, whereas SORA additionally enforces trust–risk validation, policy gating, cross-agent consistency checks, and dual-chain anchoring. Accordingly, the comparison emphasizes runtime realism rather than absolute performance parity.

Measured performance comparison. Table 9 presents a side-by-side comparison of runtime performance across increasing request volumes. For SORA, all reported values correspond to a three-agent deployment operating in a GPT-4-only configuration, without MAE-based model selection, and are obtained from direct runtime measurements on the physical testbed described in Section 7.2. Benchmark values for GPT-3 and GPT-4 are reproduced from [55] under their Python-based MultiChain configuration.

Table 9
Performance comparison: SORA (3 agents, GPT-4 only) vs. GPT-optimized smart contracts from [55].

Requests	Throughput (req/s)			Operational / Governance Delay (ms)		
	SORA (GPT-4)[a]	GPT-3[a]	GPT-4[a]	SORA (GPT-4)[a]	GPT-3[a]	GPT-4[a]
100	17.2	1.41	1.65	21	409.00	368.00
500	16.3	7.80	9.10	32	135.63	120.35
1000	15.2	14.50	17.85	52	95.63	85.22
2000	13.8	28.10	32.40	92	70.88	63.45

[a] **Notes:** SORA results correspond to a GPT-4-only configuration with governance validation and dual-chain anchoring enabled. GPT-3 and GPT-4 benchmark values are reproduced from Tables 22 and 23 of [55]. Operational delay in SORA corresponds to governance delay (D), including validation, policy enforcement, and SORA-chain anchoring.

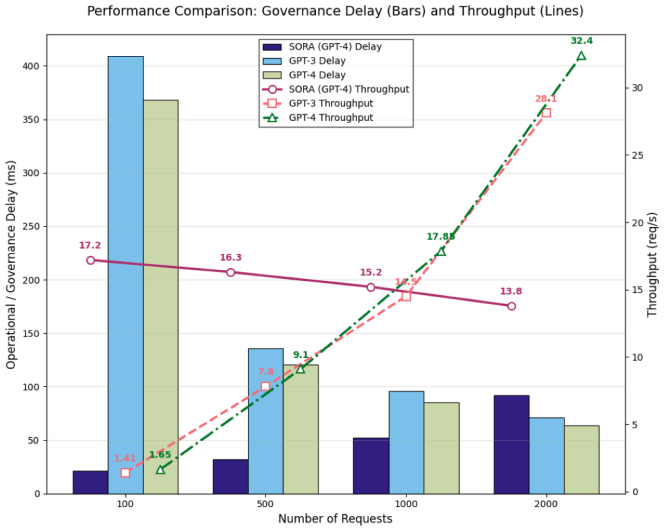

Fig. 12. Comparative runtime performance of SORA (GPT-4-only) and GPT-optimized smart contract systems [55]. Bars represent operational/governance delay, while lines indicate achieved throughput across increasing request volumes. The figure highlights that SORA maintains competitive throughput (13.8–17.2 req/s) with substantially lower governance delay despite enforcing trust–risk validation, cross-agent policy checks, and dual-chain anchoring.

Interpretation and realism of reported throughput. All SORA measurements reported in Table 9 are obtained from direct runtime measurements, with no queueing-based or analytical projection models applied in this analysis. As illustrated in Fig. 12, SORA sustains throughput in the range of 13.8–17.2 req/s while simultaneously enforcing trust–risk validation, cross-agent policy checks, and dual-chain anchoring. These values fall within the performance envelope reported by GPT-optimized smart contract systems, which span approximately 1.4–51.8 req/s depending on workload characteristics and optimization focus. Importantly, both the tabular and visual comparisons show that SORA maintains substantially lower operational delay (21–92 ms) than the benchmark system (52–409 ms), reflecting an optimized governance pipeline that preserves responsiveness despite additional enforcement responsibilities.

Architectural trade-offs and complexity perspective. From a complexity standpoint, SORA's end-to-end governance processing scales linearly with the number of agents and candidate models, i.e., $O(N|A||M|)$ per batch (Table 10), with blockchain anchoring introducing bounded constant overhead. In contrast, the benchmark system prioritizes cryptographic contract execution efficiency, achieving lower per-contract execution time at the expense of higher end-to-end operational delay. The observed throughput and delay trends are therefore consistent with the respective architectural objectives of the two systems and do not indicate unrealistic performance measurements.

Overall, this runtime validation analysis confirms that the reported throughput and governance delay of SORA are realistic for a dual-chain, multi-agent AI governance framework. The results align with independently published GPT-based blockchain systems while reflecting the additional enforcement responsibilities intrinsic to trust–risk-aware smart-city governance.

8. Complexity analysis and research question validation

This section complements the experimental results by providing an analytical assessment of the proposed framework and a synthesis of how the full paper addresses the stated research questions. The analysis combines architectural design, workflow enforcement, algorithmic governance, empirical evaluation, complexity analysis and correctness guarantees to validate the proposed hypothesis at a system level.

8.1. Analytical complexity and correctness guarantees

The proposed architecture integrates deterministic perception models, parallel LLM-based reasoning, and a governance layer with blockchain anchoring. Let $|A|$ denote the number of active domain agents, $|M|$ the number of LLMs used per agent, and N the number of requests processed within a given evaluation window. At the agent level, perception and prediction costs are dominated by domain-specific ML models such as XGBoost (for tabular weather data) and YOLO-based detectors (for traffic and safety imagery). These operations execute independently per agent and exhibit bounded, domain-dependent complexity. Prompt construction and trust–risk computation incur constant time overhead. At the governance level, each request requires evaluating $|M|$ candidate LLM outputs against policy thresholds, computing Mean Absolute Error (MAE), and enforcing security, domain, and cross-domain constraints. As a result, governance processing scales linearly with the number of candidate models and active agents. Blockchain anchoring introduces a constant per-decision overhead, which becomes the dominant factor under high request volumes.

Table 10 summarizes the asymptotic time and space complexity of each pipeline component. The linear dependence on $|M|$ and $|A|$ explains the moderate increase in execution time and governance delay observed in Section 7.2, while the absence of super-linear terms confirms the practical scalability of the framework for multi-agent smart-city deployments.

8.2. Research question validation

This subsection synthesizes evidence from the architecture, workflow execution, governance algorithms, analytical guarantees, and experimental evaluation to validate the stated research questions.

1. RQ1: Continuous validation of Agentic AI outputs. It is addressed through the multi-layer validation pipeline that combines agent-level reasoning with governance-level enforcement. Domain-specific agents process sensor data and generate trust–risk estimates using multiple LLMs. These outputs are continuously validated by SORA through MAE-based comparison against governance reference values, policy gates (S1–S3), and fallback mechanisms. The workflow in Section 6 and the convergence trends in Figs. 8 and 9 show that governance-guided feedback progressively aligns agent outputs with policy-consistent behavior while preventing unsafe decisions.
2. RQ2: Decentralized transparency and accountability via blockchain. It is addressed by the dual-chain governance design. Agent-level decisions and intermediate reasoning outputs are immutably logged on the Agentic Blockchain to ensure provenance. Final governance decisions, escalations, and cross-domain actions are anchored on the

Table 10
Asymptotic time and space complexity per decision interval t for the proposed agentic AI pipeline.

Pipeline Component	Operation Description	Time Complexity	Space Complexity
Weather perception (XGBoost)	Weather regime classification from API features	$O(T_{\text{xgb}}(d))$	$O(d)$
Traffic perception (YOLOv8)	Vehicle detection and counting	$O(T_{\text{yolo}}(I)) + O(K)$	$O(K)$
Safety perception (YOLO Fire/Smoke)	Hazard detection and confidence estimation	$O(T_{\text{yolo}}(I)) + O(K)$	$O(K)$
Prompt construction	Hard-coded template population with real values	$O(1)$	$O(1)$
Multi-LLM reasoning (per agent)	Parallel inference using GPT-4, Grok, DeepSeek	$O(\|M\|)$ API calls	$O(\|M\|)$
Agent-side trust and risk computation	Compute $R_i(t)$ and $T_i(t)$	$O(1)$	$O(1)$
Agentic-chain logging	Append signed agent output to ledger	$O(B_a)$	$O(1)$
SORA admission gate (S1)	Validate ΔR, ΔT, and trust baselines	$O(\|M\|)$	$O(\|M\|)$
MAE computation (S2)	Compute MAE for candidate LLM outputs	$O(\|M\|)$	$O(1)$
Model selection and tie-breaking	Select lowest-MAE candidate	$O(\|M\|)$	$O(1)$
Error-directed feedback (S3)	Compute ΔR and ΔT	$O(\|M\|)$	$O(\|M\|)$
Policy enforcement	Security, domain, and cross-domain checks	$O(\|P\|)$	$O(1)$
Ecosystem aggregation	Compute $T_{\text{eco}}(t)$ and $R_{\text{eco}}(t)$	$O(\|A\|)$	$O(1)$
SORA-chain anchoring	Append governance decision to ledger	$O(B_s)$	$O(1)$
Batch of N requests	End-to-end governance processing	$O\big(N(\|A\|\|M\| + \|A\|\|P\| + B_a + B_s)\big)$	$O(\|A\|\|M\|)$

SORA Blockchain for global auditability. This separation enables decentralized autonomy without sacrificing accountability. The architectural design (Figs. 2 and 3) and runtime validation results confirm that blockchain anchoring introduces bounded overhead while ensuring transparent and tamper-resistant governance.

3. RQ3: Adaptive trust regulation and orchestration under risk. It is addressed through adaptive trust–risk regulation enforced by the SORA Governance Layer. Domain-specific admission thresholds, global risk aggregation, and cross-agent coordination rules dynamically regulate agent autonomy. Under high-risk conditions, SORA restricts, escalates, or synchronizes agent actions while preserving safety invariants. Disaster-response use cases and operational results demonstrate that trust and risk thresholds effectively control escalation behavior, maintain regulatory compliance, and enable coordinated responses across heterogeneous domains.

Together, these findings validate the research hypothesis by demonstrating that continuous governance validation, adaptive trust regulation, and auditable dual-chain accountability jointly enable trustworthy and regulation-compliant Agentic AI in smart-city environments.

9. Limitations of the proposed approach

While the proposed agentic AI ecosystem introduces a principled and governance-aligned framework for smart-city decision-making, several limitations must be acknowledged to contextualize its applicability and to guide future extensions.

9.1. Residual risk of hallucination and semantic drift

Although the framework mitigates hallucination by separating deterministic perception (XGBoost, YOLO) from LLM-based reasoning, constraining prompts, and enforcing governance-level validation, residual risks remain. LLMs are probabilistic and may exhibit semantic drift under ambiguous or incomplete inputs, and the framework does not provide formal guarantees of hallucination elimination. The evaluation, therefore, focuses on trust and risk alignment with governance reference values rather than direct operational safety outcomes. Reduced trust and risk MAE indicates improved policy-aligned behavior and more reliable triggering of safety thresholds, which is a necessary precondition for safe operation. However, these metrics act as governance-level proxies for safety and do not directly quantify real-world impacts such as reduced false alarms or faster emergency response, motivating future safety-case validation and human-in-the-loop oversight.

9.2. Computational and economic cost

The use of multiple LLMs in parallel introduces non-trivial computational and monetary costs, especially when deployed at city scale or under high-frequency decision cycles. API-based inference incurs usage-dependent charges, and the cumulative cost of querying GPT-4, Grok, and DeepSeek concurrently may limit feasibility for budget-constrained municipalities. Although the framework is designed to scale horizontally and supports selective invocation strategies, cost optimization (e.g., adaptive model selection or tiered reasoning) remains an open challenge and a key direction for future work.

9.3. Rate limits and latency constraints

Dependence on external LLM APIs introduces operational constraints related to rate limits, throughput ceilings, and network-induced latency. During peak demand periods or large-scale emergencies, API throttling could delay reasoning responses and degrade system responsiveness. While the governance architecture tolerates short delays through buffering and asynchronous validation, strict real-time guarantees cannot be assured in all conditions. Mitigating these constraints may require hybrid deployments incorporating on-premise or open-weight models for fallback operation.

9.4. Robustness under adversarial manipulation

The framework assumes that upstream ML models and data sources provide reliable, integrity-checked inputs. However, adversarial manipulation remains a concern. Sensor spoofing, data poisoning, or carefully crafted perturbations in visual inputs may propagate misleading evidence into the reasoning layer. Similarly, prompt injection attacks targeting LLM APIs could attempt to influence semantic reasoning. Although cryptographic integrity checks, governance thresholds, and cross-model disagreement act as partial defenses, the framework does not yet include dedicated adversarial detection or formal verification mechanisms. Strengthening resilience against coordinated adversarial strategies is therefore a critical area for future enhancement.

9.5. Dependency on external model behavior

Finally, the framework relies on the evolving behavior of proprietary and third-party LLMs whose internal architectures, training data, and update schedules are not fully transparent. Model updates may alter reasoning characteristics over time, potentially affecting trust–risk alignment or convergence behavior. While governance feedback loops help compensate for such drift, long-term stability cannot be fully guaranteed without tighter control over model versions or standardized audit interfaces.

Overall, these limitations do not undermine the core contributions of the proposed approach but rather delineate the current boundaries of agentic AI deployment in safety-critical smart-city environments. Addressing these challenges will be essential for transitioning from

controlled experimental settings to sustained, city-scale operational adoption.

10. Conclusion and future directions

In conclusion, this work introduced *SORA-ATMAS*, a principled governance framework integrating decentralized agentic intelligence with centralized oversight and dual-chain anchoring for resilient smart-city disaster management. By embedding governance and escalation policies (S1-S6), the framework enables heterogeneous agents (Weather, Traffic, and Safety/Fire) to operate autonomously while remaining accountable to city-wide policies. Evaluation showed that multiple LLMs (GPT, Grok, DeepSeek), including SORA's fallback mechanism for high-risk scenarios, converged toward policy-aligned baselines, with domain-specific optimization (Grok in Traffic, DeepSeek in Weather/Safety) reducing trust deviation (MAE) by $\approx 35\%$ while maintaining stable risk alignment. Throughput between 13.8-17.2 req/s, execution times of 58-72 ms, and governance delays of 21-92 ms confirm real-time oversight feasibility under load. Policy-driven coordination further prevented unsafe or conflicting actions, such as traffic rerouting constrained by weather and Fire/Smoke validations. Overall, *SORA-ATMAS* provides a regulation-aligned, verifiable, and context-aware framework for transforming distributed agent outputs into accountable, city-scale decisions, establishing a resilient foundation for governance in smart-city disaster-management scenarios, with potential extensibility to broader smart-city services subject to further large-scale empirical validation.

Future research must now focus on hardening the framework against privacy risks, security threats, and adversarial conditions while further extending its scalability and applicability. Privacy-preserving analytics such as differential privacy, homomorphic encryption, and secure enclaves should be integrated to safeguard inter-domain data exchanges, complemented by selective disclosure mechanisms like zero-knowledge proofs. Strengthened cryptographic assurances through post-quantum primitives, hierarchical key rotation, and formally verified contracts will further reinforce auditability and compliance. Resilience against adversarial scenarios, including data poisoning, sensor spoofing, and LLM prompt injection, will require embedding uncertainty quantification and adversarial detection into both agent-level and supervisory layers. In parallel, static enforcement thresholds can evolve into adaptive, learning-based controllers that preserve safety invariants while adapting to context and operator input. Building on the runtime evaluation and scalability projections presented in Section 7.2, future work should incorporate large-scale stress tests beyond nine-agent simulations, exploring SORA sharding, partial asynchrony, and multichain partitioning to ensure sustained throughput and bounded governance delays in city and multi-city deployments. Although the dual-chain architecture is designed for scalability, empirical validation at large scales (e.g., 10, 50, 100+ agents) remains future work. Scaling to city wide deployments will require SORA blockchain sharding and hybrid governance mechanisms to sustain throughput, which should be evaluated through large-scale city digital-twin simulations before real-world deployment. Scaling SORA-ATMAS with federated policy learning, digital-twin based simulations, and standardized audit trails will provide the operational validation and regulatory grounding necessary for deployment. Advancing along these directions will consolidate SORA-ATMAS as a scalable, privacy-preserving, and resilient foundation for governance in next-generation smart cities.

CRediT authorship contribution statement

Usama Antuley: Writing – original draft, Visualization, Methodology, Investigation, Formal analysis, Conceptualization; **Shahbaz Siddiqui:** Writing – original draft, Visualization, Supervision, Methodology, Conceptualization; **Sufian Hameed:** Writing – review & editing, Validation, Supervision, Methodology, Conceptualization; **Waqas Arif:** Writing – original draft, Methodology, Investigation, Conceptualization; **Syed Attique Shah:** Writing – review & editing, Validation, Supervision, Funding acquisition, Conceptualization.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.knosys.2026.115403

References

[1] M.M. Rathore, A. Ahmad, A. Paul, S. Rho, Urban planning and building smart cities based on the internet of things using big data analytics, Comput. Netw. 101 (2016) 63–80.

[2] I.H. Sarker, Smart city data science: towards data-driven smart cities with open research issues, Internet Things 19 (2022) 100528.

[3] O.J. Adeleke, K. Jovanovich, S. Ogunbunmi, O. Samuel, T.O. Kehinde, Comprehensive exploration of smart cities: a systematic review of benefits, challenges, and future directions in telecommunications and urban development, IEEE Sens. Rev., 2, 228–245. 2025.

[4] J.C.N. Bittencourt, D.G. Costa, P. Portugal, F. Vasques, A survey on adaptive smart urban systems, IEEE Access, 12, 102826–102850. 2024,

[5] M. Al-Raeei, The smart future for sustainable development: artificial intelligence solutions for sustainable urbanization, Sustain. Develop. 33 (1) (2025) 508–517.

[6] A. Tiwari, Conceptualising the emergence of agentic urban ai: from automation to agency, Urban Inform. 4 (13) (2025). https://doi.org/10.1007/s44212-025-00079-7

[7] S. Murugesan, The rise of agentic ai: implications, concerns, and the path forward, IEEE Intell. Syst. 40 (2) (2025) 8–14.

[8] D.B. Acharya, K. Kuppan, B. Divya, A. Agentic, Autonomous intelligence for complex goals: a comprehensive survey, IEEE Access 13 (2025) 18912–18936. https://doi.org/10.1109/ACCESS.2025.3532853

[9] X. Wang, H. Wang, B. Bhandari, L. Cheng, AI-empowered methods for smart energy consumption: a review of load forecasting, anomaly detection and demand response, Int. J. Precis. Eng. Manufac. Green Technol. 11 (3) (2024) 963–993. https://doi.org/10.1007/s40684-023-00537-0

[10] S. Hosseini, H. Seilani, The role of agentic ai in shaping a smart future: a systematic review, Array 26 (2025) 100399. https://doi.org/10.1016/j.array.2025.100399

[11] G.C. Homsy, Z. Liu, M.E. Warner, Multilevel governance: framing the integration of top-down and bottom-up policymaking, Int. J. Pub. Admin. 42 (7) (2019) 572–582.

[12] K.M. L.D. Lartey, Artificial intelligence adoption in urban planning governance: a systematic review of advancements in decision-making, and policy making, Landsc. Urban Plan. 258 (2025) 105337. https://doi.org/10.1016/j.landurbplan.2025.105337

[13] N.A. Smuha, Regulation 2024/1689 of the eur. parl. & council of june 13, 2024 (eu artificial intelligence act), Int. Legal Mater., 64 2025, pp. 1–148.

[14] M. Homaei, O. Mogollón-Gutiérrez, J. Sancho, A review of digital twins and their application in cybersecurity based on artificial intelligence, Artif. Intell. Rev. 57 (2024). https://doi.org/10.1007/s10462-024-10805-3

[15] V.S. Rajkumar, A. Ştefanov, A. Presekal, P. Palensky, J.L.R. Torres, Cyber attacks on power grids: causes and propagation of cascading failures, IEEE Access 11 (2023) 103154–103176. https://doi.org/10.1109/ACCESS.2023.3317695

[16] T.W. Sanchez, M. Brenman, X. Ye, The ethical concerns of artificial intelligence in urban planning, J. Am. Plann. Assoc. 91 (2) (2025) 294–307.

[17] R. Islam, R. Bose, S. Roy, Decentralized trust framework for smart cities: a blockchain-enabled cybersecurity and data integrity model, Sci. Rep. 15 (2025) 23454. https://doi.org/10.1038/s41598-025-06405-y

[18] S. Musik, J. Sasin-Kurowska, M. Panczyk, Bridging the past and future of clinical data management: the transformative impact of artificial intelligence, Open Access J. Clin. Trials 17 (2025) 15–33.

[19] S. Siddiqui, S. Hameed, S.A. Shah, J. Arshad, Y. Ahmed, D. Draheim, A smart-contract-based adaptive security governance architecture for smart city service interoperations, Sustain. Cities Soc. 113 (2024) 105717. https://doi.org/10.1016/j.scs.2024.105717

[20] G. Mustafa, W. Rafiq, N. Jhamat, Z. Arshad, F.A. Rana, Blockchain-based governance models in e-government: a comprehensive framework for legal, technical, ethical and security considerations, Int. J. Law Manag. 67 (1) (2025) 37–55.

[21] S. Siddiqui, S. Hameed, S.A. Shah, A.K. Khan, A. Aneiba, Smart contract-based security architecture for collaborative services in municipal smart cities, J. Syst. Archit. 135 (2023) 102802. https://doi.org/10.1016/j.sysarc.2022.102802

[22] B. Könighofer, J. Rudolf, A. Palmisano, M. Tappler, R. Bloem, Online shielding for reinforcement learning, Innov. Syst. Softw. Eng. 19 (4) (2023) 379–394. https://doi.org/10.1007/s11334-022-00480-4
[23] T. Kanwal, A.A. Jabbar, A. Anjum, S.U. Malik, A. Khan, N. Ahmad, U. Manzoor, M.N. Shahzad, M.A. Balubaid, Privacy-aware relationship semantics-based xacml access control model for electronic health records in hybrid cloud, Int. J. Distrib. Sens. Netw. 15 (6) (2019) 1550147719846050.
[24] E.A. Odedina, Redefining governance, risk, and compliance (grc) in the digital age: integrating ai-driven risk management frameworks, World J. Adv. Eng. Technol. Sci. 10 (01) (2023) 264–282. https://doi.org/10.30574/wjaets.2023.10.1.0257
[25] M. Sadeghi, A. Carenini, O. Corcho, M. Rossi, R. Santoro, A. Vogelsang, Interoperability of heterogeneous systems of systems: from requirements to a reference architecture, J. Supercomput. 80 (7) (2024) 8954–8987.
[26] A. Huseinović, S. Mrdović, K. Bicakci, S. Uludag, A survey of denial-of-service attacks and solutions in the smart grid, IEEE Access 8 (2020) 177447–177470. https://doi.org/10.1109/ACCESS.2020.3026923
[27] G.D. Vito, F. Palomba, F. Ferrucci, The role of large language models in addressing iot challenges: a systematic literature review, Fut. Gen. Comput. Syst. 171 (2025) 107829. https://doi.org/10.1016/j.future.2025.107829
[28] B.A. Adewusi, B.I. Adekunle, S.D. Mustapha, A.C. Uzoka, A conceptual model for responsible ai integration in public-facing digital services and platform governance, Int. J. Sci. Res. Comput. Sci. Eng. Inform. Technol. 10 (2) (2024) 416–435. https://doi.org/10.32628/CSEIT2425416735
[29] U. Antuley, S. Hameed, S. Siddiqui, S.A. Shah, Securing smart city ecosystems: a taxonomy-based review of emerging technologies and frameworks for scalable collaborative services, IET Smart Cities 7 (1) (2025) 70007. https://doi.org/10.1049/smc2.70007
[30] P. Dazzi, The internet of ai agents (iaia): a new frontier in networked and distributed intelligence, Int. J. Netw. Distrib. Comput. 13 (2025) 16. https://doi.org/10.1007/s44227-025-00057-0
[31] F. Golpayegani, N. Chen, N. Afraz, E. Gyamfi, A. Malekjafarian, D. Schäfer, C. Krupitzer, Adaptation in edge computing: a review on design principles and research challenges, ACM Trans. Auton. Adapt. Syst. 19 (3) (2024) 1–43.
[32] K. Ayub, R. Alshawa, A secure iot framework for smart cities: integrating servicenow irm/grc with blockchain and ai-driven threat detection, in: 2024 International Conference on Computer and Applications (ICCA), 2024, pp. 1–6. https://doi.org/10.1109/ICCA62237.2024.10927950
[33] T. Wang, J. Cao, A. Hussain, Adaptive traffic signal control for large-scale scenario with cooperative group-based multi-agent reinforcement learning, Transp. Res. Part C Emerging Technol. 125 (2021) 103046. https://doi.org/10.1016/j.trc.2021.103046
[34] C. Cai, M. Wei, Adaptive urban traffic signal control based on enhanced deep reinforcement learning, Sci. Rep. 14 (2024) 14116. https://doi.org/10.1038/s41598-024-64885-w
[35] S. Hameed, A. Islam, K. Ahmad, Deep learning based multimodal urban air quality prediction and traffic analytics, Sci. Rep. 13 (2023) 22181. https://doi.org/10.1038/s41598-023-49296-7
[36] S. Elamanov, H. Son, B. Flynn, S.K. Yoo, N. Dilshad, J. Song, Interworking between modbus and internet of things platform for industrial services, Dig. Commun. Netw. 10 (2) (2024) 461–471. https://doi.org/10.1016/j.dcan.2022.09.013
[37] J. Lee, J. Kim, S.K. Yoo, T. Taleb, J. Song, Standardised interworking and deployment of iot and edge computing platforms, Dig. Commun. Netw. 2025. https://doi.org/10.1016/j.dcan.2025.04.006
[38] S. Praharaj, Command and control governance in the 100 smart cities mission in India: urban innovation or utopias?, Appl. Geogr. 184 (2025) 103766. https://doi.org/10.1016/j.apgeog.2025.103766
[39] A. Sohail, B. Shen, M.A. Cheema, Beyond data, towards sustainability: a sydney case study on urban digital twins„ PFG 93 (2025) 365–377. https://doi.org/10.1007/s41064-025-00337-y
[40] R. Rosmaninho, D. Raposo, P. Rito, S. Sargento, Edge-cloud continuum orchestration of critical services: a smart-city approach, 2025. https://doi.org/10.1109/TSC.2025.3568251
[41] T. Wu, P. Zhou, K. Liu, Y. Yuan, X. Wang, H. Huang, D.O. Wu, Multi-agent deep reinforcement learning for urban traffic light control in vehicular networks, IEEE Trans. Veh. Technol. 69 (8) (2020) 8243–8256. https://doi.org/10.1109/TVT.2020.2997896
[42] Z. Li, H. Yu, G. Zhang, S. Dong, C.-Z. Xu, Network-wide traffic signal control optimization using a multi-agent deep reinforcement learning, Transp. Res. Part C Emerging Technol. 125 (2021) 103059. https://doi.org/10.1016/j.trc.2021.103059
[43] D. Ozsoyeller, Öznur Özkasap, Distributed asynchronous rendezvous planning on the line for multi-agent systems, Fut. Gen. Comput. Syst. 161 (2024) 35–48. https://doi.org/10.1016/j.future.2024.06.054
[44] B. Könighofer, J. Rudolf, A. Palmisano, M. Alshiekh, E. Bartocci, R. Bloem, K. Chatterjee, T.A. Henzinger, A. Turrini, Online shielding for reinforcement learning, Innov. Syst. Softw. Eng. 19 (2023) 379–394. https://doi.org/10.1007/s11334-022-00480-4
[45] K. Riad, J. Cheng, Adaptive xacml access policies for heterogeneous distributed iot environments, Inf. Sci. 548 (2021) 135–152. https://doi.org/10.1016/j.ins.2020.09.051
[46] A. Al-Haj, Enhancing iot security schemes through governance, risk, and compliance (grc), in: S. Li (Ed.), Information Management, Springer Nature Switzerland, 2026, pp. 217–226. https://doi.org/10.1007/978-3-031-99353-4_19
[47] M.M. Karim, D.H. Van, S. Khan, Q. Qu, Y. Kholodov, Ai agents meet blockchain: a survey on secure and scalable collaboration for multi-agents, Fut. Internet 17 (2) (2025). https://doi.org/10.3390/fi17020057
[48] F. Haseeb, S. Ali, N. Ahmed, N. Alarifi, Y.M. Youssef, Comprehensive probabilistic analysis and practical implications of rainfall distribution in Pakistan, Atmosphere 16 (2) (2025) 122. https://doi.org/10.3390/atmos16020122
[49] M. Amjad, A. Khan, K. Fatima, O. Ajaz, S. Ali, K. Main, Analysis of temperature variability, trends and prediction in the karachi region of pakistan using arima models, Atmosphere 14 (1) (2022) 88. https://doi.org/10.3390/atmos14010088.
[50] F. Nekouee, Yolov8 Traffic Density Estimation, 2023 https://github.com/FarzadNekouee/YOLOv8_Traffic_Density_Estimation.
[51] A. Kausar, S. Zubair, H. Sohail, Evaluating the challenges and impacts of mixed-use neighborhoods on urban planning: an empirical study of a megacity, Karachi, Pakistan, Discover Sustain. 5 (24) (2024) 1–14. https://doi.org/10.1007/s43621-024-00195-5
[52] S. Gamal, Real-Time Smoke and Fire Detection with Yolo, 2024. https://github.com/sayedgamal99/Real-Time-Smoke-Fire-Detection-YOLO11.
[53] Artificial Intelligence Risk Management Framework (ai rmf 1.0), Functions: Govern, Map, Technical Report, National Institute of Standards and Technology (NIST), 2023. https://www.nist.gov/itl/ai-risk-management-framework.
[54] H. Chaudhary, G. Sharma, D. Nishad, S. Khalid, Ai-enhanced modelling of queueing and scheduling systems in cloud computing, Discover Appl. Sci. 7 (4) (2025) 276. https://doi.org/10.1007/s42452-025-06755-2.
[55] S. Misbah, M.F. Shahid, S. Siddiqui, T.J.S. Khanzada, R.B. Ashari, Z. Ullah, M. Jamjoom, Generative AI-driven smart contract optimization for secure and scalable smart city services. Smart Cities 8 (4) (2025) 118. https://doi.org/10.3390/smartcities8040118.